For Tandy's 100 200 and 600 Portables and 1000, 1200, 2000 and 3000 MS-DOS Computers

PCM

The Personal Computer Magazine for Tandy® Computer Users

Vol. IV No. 1
July 1986

U.S. $3
Canada $3.95

Celebration
Our Third Anniversary!

Harmony
3-D Sound on your Tandy 1000

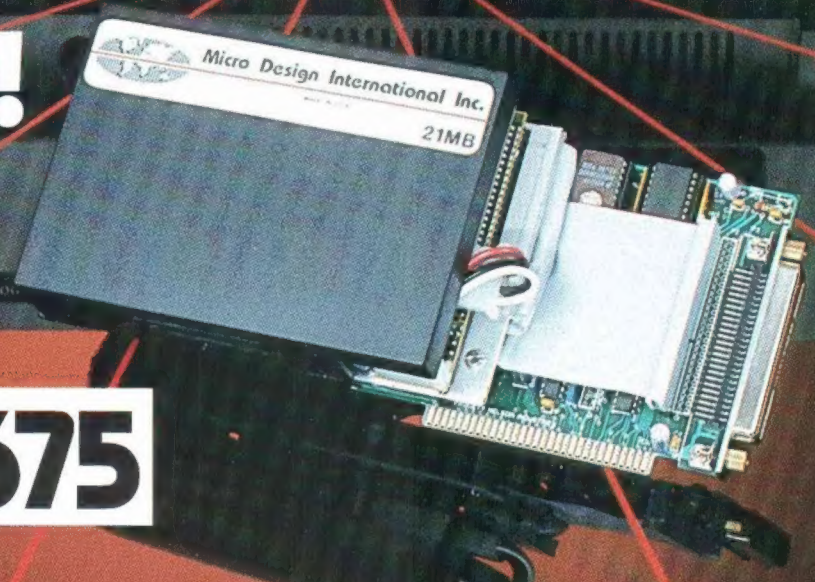
Tanyopoly
Real Estate ding Game

<PCM>>
KAREN K OKADA
4380 OGBURN AVE
WINSTON-SALEM NC 27105
413 8607
07
002

21 MEGABYTE SCSI HARD DISK CARD

NEW!!!

for your TANDY 1000



only $675

IMAGINE!!! MDI'S **21 MEGABYTE HARD DISK CARD** simply plugs into your system and is ready to run!!! The hard disk card also contains a...

"FREE SCSI PORT":

A **SCSI** port which allows you to tie (daisy chain) up to seven external (peripheral) devices to your computer.

INCLUDED **FREE**

Direct/Assist	$49.95
R/Assist	$49.95
Light Speed	$79.95
SCSI Board	$190.00

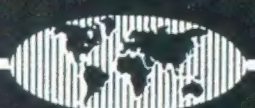
Tandy 1000 Multifunction Board $255

NOW IN OUR 8TH YEAR

ORDER LINE **800-228-0891**
305-677-8333

MasterCard VISA

1 YEAR WARRANTY, 30 DAY MONEY BACK GUARANTEE



Micro Design International, Inc.

6566 University Boulevard, Winter Park, Florida 32792 305-677-8333 telex 332559 MDII ORLUD

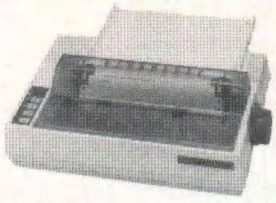
From Computer Plus to YOU . . .

PLUS *after* PLUS *after* PLUS

Tandy 200 24K $649
Model 600 32K $1269
Model 100 24K $425

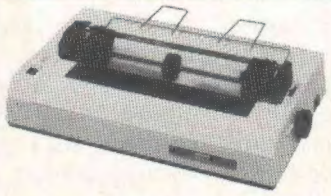
Tandy 3000 $1969
Tandy 3000 HD $2699

Tandy 1000 $685
Tandy 1000 HD $1539
Tandy 1200 HD $1599



DMP-130 $269

Color Computer II
w/64K Ext. Basic $145



DMP-105 $160

BIG SAVINGS ON A FULL COMPLEMENT OF RADIO SHACK COMPUTER PRODUCTS

COMPUTERS	
Tandy 1000 1 Drive 128K	685.00
Tandy 1000 HD 10 Meg. 256K	1539.00
Tandy 1200 10 Meg. 256K	1599.00
Tandy 3000 1 Drive 512K	1969.00
Tandy 3000 HD 20 Meg. 512K	2699.00
Model IVD 64K with Deskmate	889.00
PRINTERS	
Radio Shack DMP-130	269.00
Radio Shack DMP-430	559.00
Radio Shack DWP-220 Daisy Whee	359.00
Radio Shack TRP-100 Portable	229.00
Silver Reed EXP-550P Daisy Wheel	229.00
Star SG-10	245.00
Star SG-15	410.00
Star SD-10	365.00
Panasonic P-1091	259.00
Panasonic P-1092	339.00
Toshiba 1340	439.00
Okidata 192	375.00
Epson LX-80	275.00
Epson FX-85	419.00
Epson HS-80 Portable Ink Jet	339.00

Tandy 1000/1200 ACCESSORIES	
Tandy 1000 Disk Drive Kit	135.00
Tandy 1000 10 Meg. Hard Drive	579.00
Hard Drive Controller Board	249.00
256K Ram Board (inc. 128K & DMA)	179.00
512K Ram Board (includes 128K)	119.00
256K Memory Plus Expansion Brd.	155.00
PBJ Multi-Function Board (128K)	259.00
PBJ Multi-Function Board (256K)	279.00
PBJ Multi-Function Board (512K)	299.00
128K Ram Upgrade Kit (NEC)	78.00
256K Ram Upgrade (for PBJ Board)	69.00
300 Baud Modem Board	129.00
1200 Baud Modem Board	179.00
RS-232 Serial Board	69.00
Digi-Mouse/Clock Board	89.00
Tandy 1000/1200 MONITORS	
Tandy VM-2 Green Monitor	129.00
Tandy CM-2 RGB Color Monitor	379.00
Tandy CM-4 RGB Color Monitor	249.00
Tandy VM-3 TTL Green Monitor	179.00
Tandy 1200 Text Monitor Adapt.	219.00
Tandy 1200 Color/Graphics Adapt	210.00

MODEMS	
Radio Shack DCM-3 Modem	52.00
Radio Shack DCM-5 Modem	99.00
Radio Shack DC Modem 2212	315.00
Hayes Smartmodem II 300 Baud	169.00
Hayes Smartmodem 1200 Baud	429.00
Hayes Smartmodem 1200B	415.00
FOR EVERYONE	
Network Four Outlet Surge Prot.	69.95
CCR-81 Cassette Recorder	52.00
CCR-82 Cassette Recorder	43.00
C-20 Digital Cassette Tapes (10pk)	9.95
Verbatim SSDD Plastic Box (10pk)	24.95
Verbatim DDDD Plastic Box (10pk)	29.95
Precision SSDD Diskettes (10pk)	15.95
Precision DDDD Diskettes (10pk)	17.95
NEC 64K Ram Chips (set of 8)	39.00
NEC 256K Ram Chips (set of 8)	69.00
Flip N File Disk 50	15.00
Fanfold CleanEdge Paper (2600)	35.00

Radio Shack software 10% off.
Send for complete listing of
brand name software and hardware.

CALL TOLL FREE
1-800-343-8124

- *LOWEST POSSIBLE PRICES*
- *BEST POSSIBLE WARRANTY*
- *KNOWLEDGEABLE SALES STAFF*
- *TIMELY DELIVERY*
- *SHOPPING CONVENIENCE*

VISA AMERICAN EXPRESS MasterCard

computer plus

P.O. Box 1094
480 King Street
Littleton, MA 01460

SINCE 1973

IN MASSACHUSETTS CALL (617) 486-3193

TRS-80 is a registered trademark of Tandy Corp.

PCM

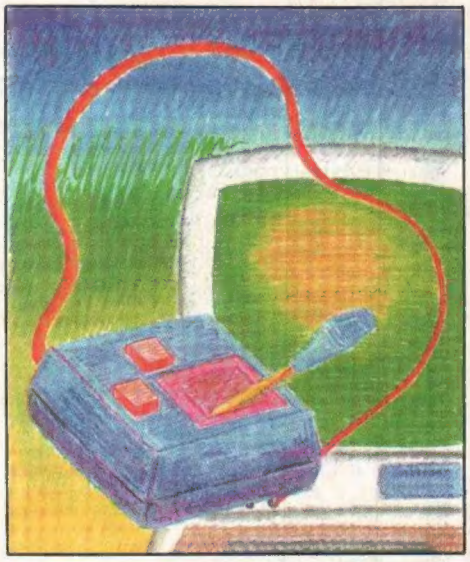
The Personal Computer Magazine
for Tandy® Computer Users

Vol. IV No. 1
July 1986

CONTENTS

Features

Departments



Reviews

Cover illustration copyright© 1986 by Fred Crawford

Tandy computers offer everything . . . almost

Check out the software that completes their systems — *filePro*®

Tandy MS-DOS Software Comparison Chart

	pfs® File/Report	dBase™ II	*filePro* 16 Smallware
GENERAL CHARACTERISTICS:			
Menu driven	yes	no	YES
Allows user to create integrated business systems	no	programmer required	YES
Developed systems and data can be moved to multi-user environments	no	no	YES
Professional support available from the software's authors	no	no	YES
PRICE	$265	$595	$495
CAPACITIES:			
Fields per record	100	32	999
Characters per record	1679	1000	4608
Records per file	1300	65535	16,000,000
Indexes per file	1	7	12
Number of digits per numeric field	20	10	24
Number of files usable concurrently	1	2	10
Files span multiple drives	no	no	up to 8
FEATURES:			
Full-screen facility for creating custom screen layouts	yes	no	YES
Full-screen facility for creating custom report layouts	no	no	YES
Built-in field types (error checking)	no	3	12
User-defined field types	no	programmer required	200
Conditional math	no	programmer required	YES
User-defined menus	no	programmer required	YES
Change file layout without losing existing data	possible	possible	automatic
Data protection	no	programmer required	YES
Password security	no	programmer required	YES

The new Tandy computers can really speed things up. If you don't let obsolete software bog you down.

Try *filePro* 16, the latest generation of data base software from the authors of Profile®.

Choose it for its power (check our capacities); its ease-of-use (check our features); and its unique ability to grow as your needs grow, and even move the systems you create to multi-user computers.

filePro is available for the Tandy 1000, 1200 and 2000 computers (hard disk required). Available directly from us or through your Radio Shack store's Express Order Software service.

To order, call (800) 847-4740.

For customer service or when dialing in New York State, call **(212) 398-9290.** Ask for Smallware sales. Visa and MasterCard accepted.

Also available for other computers, including IBM PC, IBM PC-XT, and NCR Decision Mate.

The Small Computer Company, Inc.

✓240 **230 West 41st Street, Suite 1200, New York, New York 10036**

Smallware and *filePro* are registered trademarks of The Small Computer Company, Inc. dBase II is a trademark of Ashton-Tate. pfs File/Report is a registered trademark of Software Publishing. Profile is a registered trademark of Tandy Corporation.

PCM

The Personal Computer Magazine for Tandy® Computer Users

July 1986 Vol. IV No. 1

Editor and Publisher
Lawrence C. Falk

Managing Editor Danny Humphress
Asst. Managing Editor Belinda C. Kirby
Senior Editor Tamara Renee Dunn
Submissions Editor Jutta Kapfhammer
Reviews Editor Judi Hutchinson
Editorial Assistants Judy Brashear, Wendy Falk, Angela Kapfhammer, Shirley Morgan
Technical Editor Ed Ellers
Technical Assistants Cray Augsburg, Chris Wehner
Contributing Editors William Barden, Jr., John B. Harrell III
Consulting Editors Jo Anna Arnott, T. Kevin Nickols

Art Director Tracey Jones
Designers Jody Gilbert, Heidi Maxedon, Kevin Quiggins, Sandra Underwood
Production Assistant Cindy Jett

Chief of Typography Debbie Hartley
Typography Services Jody Doyle, Suzanne Benish Kurowsky

Falsoft, Inc.

President Lawrence C. Falk

General Manager Patricia H. Hirsch
Asst. General Mgr. for Finance Donna Shuck
Admin. Asst. to the Publisher Sue E. Rodgers

Editorial Director James E. Reed
Asst. Editorial Director Jutta Kapfhammer
Creative Director Jerry McKiernan

Chief Bookkeeper Diane Moore
Advertising Accounts Beverly Taylor
Dealer Accounts Judy Quashnock
Accounts Payable/ADP Lisa Ragan
Director of Fulfillment Services Bonnie Frowenfeld
Asst. Dir. of Fulfillment Services Sandy Apple
Asst. Customer Service Manager Beverly Bearden
Word Processor Manager Patricia Eaton
Chief of Printing Services Melba Smith
Pre-press Production John Pike
Dispatch Janice Eastburn
Asst. Dispatch Mark Herndon
Business Assistants Laurie Falk, Sharon Smith, Monica Wheat, Pam Workhoven

Advertising Coordinator Doris Taylor
Advertising Representative Kim Vincent
Advertising Assistant Debbie Baxter
(502) 228-4492
West Coast Advertising and Marketing Office
Director Cindy Shackleford
Advertising Representative Shirley Duranseau
For PCM Advertising and Marketing Office Information, see Page 130

PCM — The Personal Computer Magazine for Tandy® Computer Users is published every month of the year by FALSOFT, INC., The Falsoft Building, P.O. Box 385, Prospect, KY, 40059. Phone (502) 228-4492. **PCM — The Personal Computer Magazine for Tandy® Computer Users** and the **PCM** logotypes are registered ® trademarks of FALSOFT, Inc.

Second class postage paid Prospect, KY, and additional offices. USPS N 713-470 (ISSN 0747-0460). POSTMASTER: Send address changes to PCM, P.O. Box 385, Prospect, KY 40059. Forwarding Postage Guaranteed.

Entire contents copyright © 1986 by FALSOFT, Inc. **PCM — The Personal Computer Magazine for Tandy® Computer Users** is intended for the private use and pleasure of its subscribers and purchasers and reproduction by any means is prohibited. Use of information herein is for the single end use of purchasers and any other use is expressly prohibited. All programs herein are distributed in an "as is" basis, without warranty of any kind whatsoever.

TRS-80, Model 100, Tandy 200, 600, 1000, 1200 HD, 2000, 3000 and Portable Computer are registered ® trademarks of the Tandy Corp. CompuServe is a registered ® trademark of CompuServe Inc.

Subscriptions to **PCM — The Personal Computer Magazine for Tandy® Computer Users** are $28 per year in the United States. Canadian rates are U.S. $35. Surface mail to other countries is U.S. $64, air mail U.S. $85. All subscriptions begin with the next available issue.

Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in United States currency only. Full refund after mailing of one issue. After two issues have been mailed, 10/12ths of the subscription price will be refunded. NO refund after mailing of three or more magazines.

Limited back issues are available. Please see notice for issues that are in print and their costs. Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in U.S. currency only.

Letters to PCM

New BIOS Solves Hard Disk Problems

Editor:

I recently decided to upgrade my Tandy 1000 with a hard disk as described in your magazine ("Upgrade Your Tandy 1000" February 1986). I first tried to order a 20-megabyte unit from several of the mail-order houses. To my disappointment, few said they had a unit for the 1000 and even fewer said they could deliver in a timely fashion. I ended up purchasing and returning a unit from one of your advertisers. The reason I returned it was I could not get the system to boot from the hard drive, and when writing to it from A: or B:, I received FCB errors. I called the vendor and received very courteous help and all indications pointed to correct procedures. They didn't understand why I was having problems. Disappointed, I finally ended up sending it back for a refund.

I was determined to find out why I had these problems with such a simple procedure. I talked with the Western Digital people and the Tandy folks and found out my 1000 has the old ROM BIOS version 2.11.00 (my machine was one of the first ones sold) and that there is a new version, 2.11.01, and that the hard drive doesn't have a problem using it. I called the vendor back and asked what version BIOS they were using and found they were using 2.11.01. Thought your readers might like to know to check the BIOS version number before trying to install a hard disk. With the new BIOS, all things point to being able to use just about any drive on the market provided the controller fits the slot.

Bill Kennedy
Grafton, VA

Canadian Club

Editor:

The Edmonton, Alberta, Canada Tandy users are pleased to announce the formation of a new club called the Tandy 1000 Plus Users Association. The club meets on the second Monday of every month at 12207 124 Street, Edmonton, at 7:00 PM. For more information, please phone Dexter B. Dombro at (403) 439-5245. The club also maintains its own, separate conference section on Satellite RBBS, dial (403) 474-5262. The conference deals exclusively with matters of concern to Tandy users.

The purpose of the club is to give all users a forum in which to learn and discuss issues that commonly affect Tandy users. The association is therefore open to all 1000, 1200, 2000 and 3000 users. The club supports a limited freeware library, has a technical advisor and sports a lot of enthusiasm. First time visitors to the club are admitted free.

Dexter B. Dombro, President
Tandy 1000 Plus Users Association
Edmonton, Alberta

Help for *filePro* Pros

Editor:

I own a Tandy 2000 with two floppy disk drives and use it primarily with the *filePro 16* database management system.

The 2000 is a very powerful computer and, in my mind, surpasses the IBM PC AT.

I also believe *filePro 16* outperforms *dBase III* and *RB 5000*.

I am computerizing a dental laboratory with very succesful results. However, I have had to do a lot of "trial and error" operations.

Please tell me where I can get some help with *filePro 16*.

Hartley Grim
Rockville Centre, NY

The Small Computer Company, Inc., publishes Smalltalk, *a newsletter for users of* filePro 16. *They can be reached by writing 230 West 41st Street, New York, NY 10036 or calling (212) 398-9290.*

It Works!

Editor:

In regard to a letter in the June Issue ("Fontastic Printing"), Tom says, "Borderlund's *The Printshop* does not do the trick." I thought the same when I first booted up *The Printshop* — no printer recognition. I went to sleep thinking I'd be returning the package (over 100 miles, round trip). I awoke remembering my DMP-130 has a mode that emulates IBM. I tried *The Printshop* again after setting switches 1-1 and 1-3 to on. Success!

J. Stahl
Ojo Sarco, NM

Discover a Whole New World of TV Entertainment!

SATELLITE SYSTEM

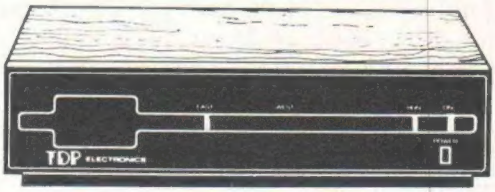
U.L. Approved

85° LNA

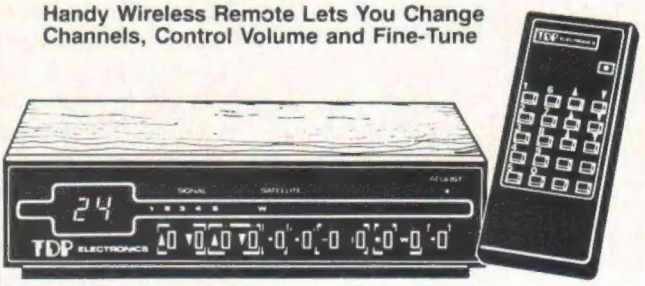
User Installable! Over 100 Channels! Uncut Movies! Specials! Sports! News! Weather! Music Videos!

Entire System **$1519**

- Totally Automatic System With 8½ -Ft. Dish, Receiver and Wireless Remote Control
- Comes With All Accessories Needed for Complete Do-it-Yourself Installation
- Descrambler Ready



Handy Wireless Remote Lets You Change Channels, Control Volume and Fine-Tune



Now you can enjoy it all! 24-hour non-stop entertainment that you simply can't get from local TV, cable, and even video tape! You'll see the hottest stars, recent uncut and uninterrupted movies, the ultimate in sports coverage—and the best part is, there are no monthly fees to pay. You'll see exciting family-oriented programs, plus music videos, FM broadcasts and movies with provision for stereo, and 24-hour news coverage on both TV and radio. It all adds up to out-of-this-world TV viewing at a down-to-earth price. Come in today and discover how easy it is to enjoy satellite programming! Includes complete installation manual.

(817) 825-4027

NOCONA ELECTRONICS • Box 593 • Nocona, TX 76255

UPS Shipping

Of Anniversaries

I think a third anniversary marks a sort of turning point for a magazine. And, since this is PCM's Third Anniversary Issue, I hope you will bear with me while my thoughts meander somewhat.

We've been in the computer magazine business here at Falsoft for five years now. This month marks PCM's third year, but also marks the fifth year for our first publication, THE RAINBOW, which covers Tandy's Color Computer. It is an auspicious month, for sure.

Let's look at anniversaries. Not that stuff about paper, crystal, diamond and so on, but just what anniversaries mean to publications.

The first year, to my mind, means you have weathered the storms of starting out and, assuming that the publication is a healthy one, means you are probably going to be around for a while. It is a time of real celebration — when you quit being a "new kid" on the block and become something more.

The second year is sort of anticlimactic, as far as I am concerned. You have dealt with all the problems engendered in the startup and it is time to define your image. That's what we did with PCM — getting into MS-DOS as well as the portables. Remember, PCM started life as a publication that supported one machine — Tandy's portable Model 100.

Three years means something else again. It is a maturing time, and a major sense of accomplishment. During the time leading up to this third anniversary we have pretty much become *the* authority for the MS-DOS machines in the Tandy stable, while still being a major force in the portable area.

We've done this simply because, with your support, we've been able to grow and mature. We've added pages. We've added features. We've added top-flight writers. In truth, we've become the source of information on Tandy's computers with the exception of their TRS-DOS, Xenix and Color computers (and THE RAINBOW takes care of the CoCo).

We expect to continue to grow. And we hope to continue to generate your support. I can't tell you how many times an advertiser has told me that he or she is continuing an ad contract because "your readers really support you — and me." Thank you. That sort of help, telling people you saw their ads here (or asking them why they don't advertise here), means we can afford to produce a bigger, better PCM for everyone!

I know I've been tooting our own horn. But none of it would be possible without two groups of people. The first, of course, is you — our reader — who does so much to make life easier for us. Your letters of encouragement are welcome, your program submissions are our bread and butter, and your support of our advertisers is heartening.

Secondly, it would be impossible to mention PCM without giving their due to the fine people at Tandy who have brought us these excellent machines. The Tandy MS-DOS computers are the finest on the market, and at the best price available. The portables are the shining examples in their price category. Where would thousands of us be without them?

So, happy birthday, PCM! We're happy and growing and we look forward to you being with us in the years ahead!

— Lonnie Falk

Each and every month, PCM offers an assortment of programs for the home and office — utilities, business applications, games and more — all for the price of typing in the listings.

If you're a person on the move, and too busy to spend time typing lengthy listings, then PCM has just what you need — **PCM ON DISK.**

PCM ON DISK provides each issue's MS-DOS programs ready to load and run with only a few keystrokes!

AS AN ADDED BONUS, **PCM ON DISK** will include PCM's portable computer programs, which may be downloaded to portables for those readers who have both an MS-DOS machine and one of Tandy's portable computers.

$99 for a year's subscription to PCM ON DISK!

That means you get a disk each month for a year, containing all programs listed in that month's issue, for an average cost of only $8.25!* We think it's worth your time to subscribe.

*Disks for individual issues (beginning with January 1986) may be ordered for only $12 each.

PCM

PCM ON DISK
The Falsoft Building
P.O. Box 385
Prospect, KY 40059
(502) 228-4492

To order by phone (*credit card orders only*) call (800) 847-0309, 8 a.m. to 5 p.m. EST. For other inquiries call (502) 228-4492.

☐ Full Year's Subscription (specify starting month) _____

☐ Single Volume (specify month(s) & year) _____

Name _____

Address _____

City _____ State _____ ZIP _____

Payment Enclosed ☐

Charge: Visa ☐ MasterCard ☐ American Express ☐

My Account# _____ Card Exp. Date _____

Signature _____

*Subscriptions to PCM ON DISK are $99 a year in the United States. Canadian rate U.S. $115. All other countries U.S. $130; PCM ON DISK single copy rate is: $12 within the U.S.; U.S. $14, Canadian rate; U.S. $16, all other countries. Kentucky residents add 5% sales tax. All subscriptions begin with the current issue unless otherwise specified. Please allow 6 to 8 weeks for first copy. PCM ON DISK is not a stand-alone product. PCM magazine is a separate purchase. **U.S. currency *only*, please.** In order to hold down non-editorial costs, we do not bill.

Wrapping Up the Workspace

Although uploading files for submission to the MS-DOS SIG database is probably the most important use of the personal Workspace area on Delphi, it is by no means the only one. The Workspace is connected, not just to the database, but to almost every other area of the SIG. This month, let's look at a few handy operations that can be performed on files that reside in the Workspace.

• The Delphi Mail feature is more closely connected to the Workspace than any other area. Every personal message you receive is stored there, lumped together in a file with a long, strange-looking filename and an .MAI extension. You'll recognize it when you see it. When you begin filing mail messages with the FILE command, they, too, are stored in the Workspace in separate "folders."

But this is just storage, and the files are readily accessible only from within Mail, not the Workspace. In other words, they have to be there; nothing handy at all about that. Let's look at some useful things we can do with a text file that's in the Workspace.

We've already discussed three ways to upload a file. When you receive Mail, you can compose your responses offline, then upload them to the Workspace. Then, when in Mail, type SEND <filename> (as it is in your Workspace) and the file is grabbed from the Workspace and mailed — very handy for sending the same message to several people, or for a stock reponse to the same questions asked by several people.

Another handy feature is the "mail distribution" file. For instance, you can create a distribution file containing the names of all your friends, or perhaps everyone you know who uses the same terminal program you do. The usernames are listed, one to a line, and the file must be named with a .DIS extension. Type CREATE FRIENDS .DIS, enter the usernames you want included and close the file with a ^Z.

Now, if you have a letter you want to distribute widely to your friends, or a question about your term program, you can send the message to them all at once. In Mail, SEND the message, and at the "To:" prompt, enter @FRIENDS, or whatever you have named your .DIS file.

Now, in the other direction, you can also save Mail message in your Workspace. For this, type EXTRACT <filename> and the text file is copied into the Workspace. Surprisingly enough, even binary program files can be sent through Mail and downloaded in this manner. That's right, you can share your programs with others without ever making them public. After starting to read the binary file, type EXTRACT <filename>/NOHEADER. This preserves the file intact, without the Mail "header," and it can then be downloaded with either Xmodem or Kermit from the Workspace.

• The Conference is connected to the Workspace by the /LOG command. When you're in a conference, you can save a transcript of the entire conversation merely by typing /LOG <filename>. The transcript is saved in your Workspace, terminating when you exit the conference or enter /NOLOG.

This is certainly a handy feature, allowing you to save entire conversations for later review in the Workspace with the LIST command, or to mail to others to read, or even to post in the Forum. All you have to do is remember the filename.

• The Forum is closely connected to the Workspace for both input and storage. Again, you can create your reponses to Forum messages offline, conserving your online time, and upload those responses into your Workspace. In the Forum, type ADD <filename> and respond to the prompts that follow. Your message is pulled from your Workspace and posted.

Conversely, if you're reading a Forum message that interests you, type FILE <filename> and that message is stored away in your Workspace where you won't lose track of it.

Unfortunately, due to a bug in the current version of the Forum software, it is not possible to reply to a message with REP <filename>. But this should be fixed very soon.

Though there are many other things you can do in your Workspace and with the files stored there, these are a few of the most useful. Next month we'll move on out into the database and investigate how to download some of the fantastic programs available in the MS-DOS SIG.

Kevin Nickols
MS-DOS SIG Manager

TANDY IS DANDY...

...until you compare

	Tandy Memory Plus	Zuckerboard Expansion Memory
DMA	Inc.	Inc.
with 256K	$319.95	$109
with 512K	$519.90	$149
Options:		
Clock/Calendar	$ 99.95*	$49
256K Upgrade	$199.95	$49
RAM Disk/ Print Spooler	N/A	$49
Warranty	90 Days *with mouse	2 Years

MADE IN U.S.A.

INTRODUCTORY OFFER PCMA

*Send the **Zuckerboards** I need for my TANDY 1000 right away.*
Here's my check ☐ or money order ☐ **MADE PAYABLE TO A.T.D.**

____ EM with 256K	@ $109 = ____	Subtotal	____
____ EM with 512K	@ $149 = ____	CA residents	
____ Clock/calendar	@ $ 49 = ____	add 7% tax	____
____ 256K upgrade	@ $ 49 = ____	Shipping	$6.00
____ RAM Disk/ Print spooler	@ $ 49 = ____	**Amount Enclosed**	____
____ Multifunction Board	@ $249 = ____		
____ 10 MB Hard Disk	@ $529 = ____		
____ 20 MB Hard Disk	@ $599 = ____		

NAME ____

ADDRESS ____

CITY ____ STATE ____ ZIP ____

TELEPHONE( ) ____

A.T.D. 1287 Lawrence Station Rd., Sunnyvale, CA 94089

NEW!!

ZUCKERBOARD PRODUCTS FOR THE TANDY 1000

Multifunction Board **$249**
Includes DMA controller chip, 512K memory, Clock/calendar with 20 year battery, Serial port, and Ramdisk/Print spooler software.

Internal Hard Disk **10 Mbyte** **$529**
20 Mbyte **$599**
Half-height. Includes controller and cables. Mounts internally, above floppy drive A. Requires system with DMA already installed.

CALL TOLL-FREE TO ORDER
CA (800) 233-6874
(800) 222-4920

Dr. Dr. Zucker

It's another ZUCKERBOARD

ATD 1287 Lawrence Station Road • Sunnyvale, CA 94089 • (408) 734-4631

ZUCKERBOARD is a Trademark of Advanced Transducer Devices, Inc.
TANDY 1000 is a Trademark of Radio Shack, a Division of Tandy Corporation.

All prices subject to change without notice due to fluctuations in the chip market.

Cursor Conversions

By Dr. Leo Finkelstein, Jr.

Converting portable BASIC programs to GW-BASIC just got a little simpler

One of the miseries I've found in writing programs in BASIC on my Tandy 200, and later converting them to the Tandy 1000, or vice versa, is that the 200 uses PRINT@ statements to locate the cursor, while GW-BASIC uses the LOCATE statement. I wrote this short program to quickly convert values from one type of statement to the other. It's designed to run on either the Tandy 200, or in GW-BASIC. In the case of GW-BASIC, it's designed for use with medium graphics resolution and a text width of 40 columns (modes 1 or 4 on the Tandy 1000). □

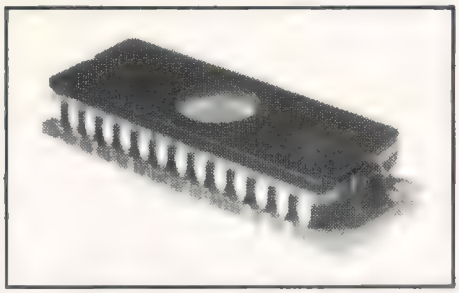
Leo Finkelstein, Jr. is currently a lieutenant colonel in the Air Force Space Command at Colorado Springs, as well as an adjunct associate professor of communication at the University of Colorado.

The listing:

```
1Ø 'CONVERSION FROM PRINT@ TO LOCATE
2Ø 'BY DR. LEO FINKELSTEIN, JR.
3Ø 'WRITTEN 3/86
4Ø 'WILL RUN ON TANDY 2ØØ AND IN GW BASI
C
5Ø 'CONVERTS TANDY 2ØØ PRINT@ VALUES TO
4Ø COLUMN SCREEN MODE LOCATE VALUES
6Ø CLS
7Ø PRINT "VALUE TO INPUT IS FROM WHAT?"
8Ø PRINT:PRINT "(A) PRINT@ STATEMENT"
9Ø PRINT "(B) LOCATE STATEMENT"
1ØØ A$=INKEY$:IF A$="" THEN 1ØØ
11Ø IF A$="A" THEN 14Ø
12Ø IF A$="B" THEN 21Ø
13Ø GOTO 1ØØ
14Ø 'DETERMINE LOCATE VALUES
15Ø PRINT:INPUT "ENTER PRINT@ VALUE";P
16Ø R = INT(P/4Ø)+1:'DETERMINE ROW
17Ø CC = P/4Ø-INT(P/4Ø)
18Ø C = INT(4Ø*CC)
19Ø PRINT:PRINT "PRINT@"P" EQUALS LOCATE
"R","C
2ØØ GOTO 27Ø
21Ø 'DETERMINE PRINT@ VALUE
22Ø PRINT:PRINT "ENTER LOCATE VALUE WITH
 COMMA-E.G. 3,4"
23Ø INPUT "LOCATE X,Y";R,C
24Ø P = (R-1) * 4Ø + C
25Ø PRINT:PRINT "LOCATE"R","C"EQUALS PRI
NT@"P
26Ø GOTO 27Ø
27Ø 'DETERMINE NEXT REQUEST
28Ø PRINT:PRINT:PRINT "HIT [X] TO DO ANO
THER"
29Ø PRINT "HIT ANY OTHER KEY TO QUIT"
3ØØ A$=INKEY$: IF A$="" THEN 3ØØ
31Ø IF A$="X" THEN 32Ø ELSE END
32Ø GOTO 6Ø
```

PCM

Model 100



ROMDSK. 8K or 32K CUSTOM expansion rom for the TRS-80 Model 100. Custom programmed with YOUR programs.
8K . . . . . **$49.95** 32K **$69.95**
Call or Write **ROSS COMPUTER SERVICES**
For Ordering Information

TANDY 1000, 1200, 3000

INNOVATIVE Software

The SMART Software System, from Innovative Software.

The SMART DATA MANAGER	**$495.00**
The SMART WORD PROCESSOR	**$295.00**
The SMART SPREADSHEET (With Graphics)	**$395.00**
The SMART SPELLCHECKER	**$139.00**
The SMART SYSTEM* . . . . . **List $895.00**	**Call**

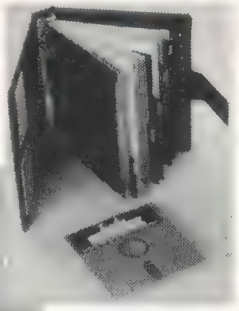
*The SMART Software System includes: The Smart Data Manager, The Smart Spreadsheet with Graphics, and The Smart Word Processor, and can only be sold as a system, not as individual products. The smart Software System includes built in asynchronous communications and personal time management programs.

SHIFT YOUR HARD DISK BACK INTO HIGH GEAR

Tune Your Disk Files For Highest Performance

with disk optimizer

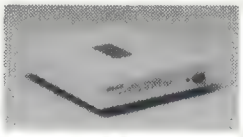
Only **$49.95**



BORLAND INTERNATIONAL

Turbo Tutor	**$ 34.95**	Superkey	**$ 69.96**
Turbo Pascal 3.0	**$ 69.95**	SideKick (copy-protected)	**$ 54.95**
Turbo Pascal with 8087	**$109.90**	SideKick (NOT copy-protected)	**$ 84.95**
Turbo Pascal with BCD	**$109.90**	Traveling SideKick (now available)	**$ 69.95**
Turbo Pascal 3.0 with both 8087 & BCD	**$124.95**	Traveling SideKick with SideKick (NCP)	**$125.00**
Turbo Database Toolbox	**$ 54.95**	Turbo Lightning	**$ 99.95**
Turbo Graphix Toolbox	**$ 54.95**	Reflex: The Analyst	**$ 99.95**
Turbo Editor Toolbox	**$ 69.95**		
Turbo Gameworks	**$ 69.95**		

MERCURY 300/ 1200 MODEM



Mercury 300/1200 auto-answer/auto-dial, HAYES compatible MODEM. Uses the industry standard AT command set.
SPECIAL **$179.95**

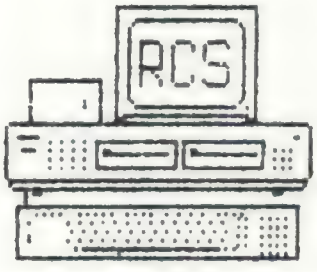
NOW JUST $49.95

ORTHO'S COMPUTERIZED GARDENING

PUT A COMPUTER TO WORK IN YOUR GARDEN!

- Regionalized to your zip code
- Over 750 plants
- You choose by common name, botanical name, height, color, soil condition, or water requirements

*Also included—"Gardening Techniques," a 196 page book with full color photography and illustrations.



ROSS COMPUTER SERVICES

(816) 363-2545

VISA MasterCard

Hours: 8:30 to 5:00
After 5:00 you will reach our order line BBS

1800 W. 91st Place • Kansas City, MO 64114

Amortization and Mail Call Tips

By Bobby Ballard

Tips and Tricks from reader mail

The help and mail has been pouring in since the last mail call, "Tips, Tricks and Letters" (April 1986). I've received lots of great tips and information from a number of readers and I want to share them with you this month.

The letters I've received are a great example of how we all benefit from sharing our experiences, solutions and questions. If you've found a trick or solution while using *DeskMate*, no matter how small it may seem to you, send it in. You never know who may be trying to solve a problem you've already licked. Look at the great responses and information that follow for an example of how we all benefit from this sharing.

Bobby Ballard is a free-lance writer and the owner of a computer software and consulting firm. He also operates a BBS in Brooklyn. Bobby can be contacted at 1207 Eighth Avenue, Apt. 4R, Brooklyn, NY 11215.

I am happy, first of all, to report that some of the information I gave you in the last mail column was in error. Happy because it means that some things I thought were impossible to do are, in reality, quite possible.

Amortizing and 23

When I last wrote on this subject, I was informed by Tandy Tower personnel that using the spreadsheet example in the tutorial to amortize loans over 23 periods or more was impossible. I was told this was a limitation of *DeskMate* and no fixes were planned. I tried to redesign the spreadsheet and still couldn't get it to work.

The good news and truth is that you can amortize loans of more than 23 periods when you use the right formula and redesign the sheet too. I received a number of responses to this problem including a copy of a letter

from inside Tandy's Towers. Duane Palmer of Carmi, Illinois, wrote directly to the chairman of Tandy, John Roach, and received a polite response containing the answer from one of Tandy's Customer Relations people, Bob Archenhold.

Bob writes, "To calculate more than 23 periods for your amortization schedule, use the following formula: #R1C2*#R2C2/(1-1/EXP(#R3C2*2.3026*LOG(1+#R2C2))). This formula uses logarithms to raise a number to a power, with 2.3026 being the conversion factor between common and natural logarithms.

"The following formula raises number 'X' to power 'Y': Result = EXP(2.3026*Y*LOG(X)). Please note that, when using this formula, you must create cells in columns long enough to hold the required number of periods. For example: to figure 48 periods, you have to extend the selection for the formulas to 48 cells down. You will see that the example in the manual says 'Select from R8-R19'. You will have to change R19 to R56, or else you will get a *0ERROR* message."

One point not made entirely clear is where to use the first formula. As Mr. Palmer points out in his accompanying letter, it is to go in payment column three (R8C3—R19C3). The second point is also important, as we've discussed here before; the columns must be increased in length to handle the new periods. To design the spreadsheet to handle 60 periods (five years), extend the cells from R19 down to R67.

I also heard from others. Michael Mazquiaran of Miami wrote with a solution that others presented too. While slightly different from the previous solution, it is no less effective. Mr. Mazquiaran presents the following formula: #R1C2*R2C2/(1-(1+#R2C2)!-#R3C2), which is represented by:

$$P = L \times \frac{i}{1-(1+i)^{-n}}$$

Mr. Mazquiaran gives credit for this formula to the Radio Shack Financial Calculator (EC-4051), page 30. The following key applies to this formula: 'P' equals Monthly payment, 'L' equals Original loan amount, 'i' equals Interest rate (on a monthly basis), and 'n' equals Number of months (periods) of the loan.

Mr. Mazquiaran is using his Tandy 1000 to accomplish some powerful tasks, keeping thousands of records on disk using Filer. And while he's moved on to other more powerful applications for some task, he is still putting *DeskMate* to work part time. He also wonders if anyone can help him track down the formula for determining the interest rate on a loan schedule when all other variables are known. He's got the other equations already worked out. Can anyone help?

Two other gentlemen wrote presenting basically the same formula that Mr. Mazquiaran presents above. Howard F. Rulf and Edward Bible, both from Wisconsin, sent in the following variations. First, Mr. Bible adds a set of parentheses to solve the problem: #R1C2*#R2C2/(1-(1/(1+#R2C2))!#R3C2) and goes on to present the following as a variation: #R1C2*R2C2/(1-(1+#R2C2)!(-R3C2)) that works equally well. Mr. Rulf though solves the problem in a different way with the same results. He sent in the following: #R1C2*#R2C2/(1-10!(-#R3C2*LOG(1+#R2C2))) to be used in R8C3 of the example under discussion. While others wrote in with solutions, all were the same as these examples or close variations. I thank everyone who wrote in with solutions.

As you can see, there are a number of solutions available to you. It's a matter of making your choice and plugging in the formula. I hope this has been of help to some of you who may have despaired at the prospect of not even being able to amortize a simple car loan, for example. I would love to hear from anyone with any further thoughts on these formulas and solutions. I think the error in the manual is possibly a typo, but, I wonder which one was intended to be included?

To help you choose which formula to use, I did some testing and have included the results in Table 1. I timed the calculation time using each of the formulas and took note of any variations in the results. The formula presented by Bob Archenhold for Tandy computes a slightly different result than the others. It is very small and of little consequence for most applications. This same formula, while the longest, tied for first in speed. The other fastest formula is the unique one presented by Howard F. Rulf. Take a look at the results in Table 1.

The Test

I like the equation presented by Mr. Rulf, personally, because it is as fast as the Radio Shack entry, returns results exactly like two of the other equations and uses less memory. The test I performed was very simple. I designed a table to amortize a loan over 60 periods and set up the interest cell to handle up to six digits. I then tracked the calculation time to amortize $100,000 at 9.25 percent annual financing rate for five years (60 periods.) The times given in Table I are from the time the last prompt, ?PERIODS, is answered at the start of the first screen write and are given in minutes. Someone, after seeing the differences, might know why they occur. My first guess would be that using the LOG function (common to both fast formulas) has an effect on speed.

Of course, this was not a contest for the fastest, just a search for a solution. I'd say the solution is found, thanks to the readers. I just wanted to see which was fastest, if any, since I noticed it took quite a while to calculate my test example. It's nice to satisfy a little curiosity now and then.

Mail Merge Too

I am also happy to report that there is a way to do a mail merge with *DeskMate*, but you must purchase additional software. While this might not be the greatest news, I must admit the inexpensive software offered to solve this problem is better than having to buy and learn a whole new package just to get mail merge capability.

Table 1

#	Formula	Time (min)	Comment
1	#R1C2*#R2C2/(1-(1+#R2C2)!-#R3C2)	2:58.3	Short
2	#R1C2*#R2C2/(1-1/EXP(#R3C2*2.3026*LOG(1+#R2C2)))	2:25.2	Memory hog
3	#R1C2*#R2C2/(1-(1/(1+#R2C2))!#R3C2)	2:30.8	Pretty fast
4	#R1C2*#R2C2/(1-10!(-#R3C2*LOG(1+#R2C2)))	2:25.2	My favorite

As a result of the first mail call column, I received notices of two new software packages that are geared toward mail merging with *DeskMate* files. One of the new programs, *CorMail*, already available on the market, is being reviewed by PCM. Keep a watch for the review if this is a capability you're interested in using with *DeskMate*. I've yet to see *CorMail*, but a copy is on the way and I will give you more details later.

I also heard from another gentleman who is supposedly working on a mail merge program for *DeskMate*. Since this product has not been completed or released, I will not print the details right now. If and when this product becomes available, I will give you the details and other information right here in "The Integrated Desk."

The first package, *CorMail* from Corwin Software, was written by William C. Corwin, Ph.D., and can be used to generate a mail merge or to print labels, show and suppress telephone numbers, print multiple labels, delete selected records and a few other features. The price is $30 as of this writing and requires GW-BASIC on a Tandy 1000 or 2000 and works with only one floppy drive. However, the author feels it might run well on the 3000 and 6000 too. You might call to find out more about those machines.

The flyer sent to me said, "The program, *CORMAIL.BAS*, accesses files in *DeskMate* to print mailing labels and to print a sequence of letters using a letter, CHNLTR.DOC, from Text and inserting an address from CORWIN.FIL in the Filer automatically without any operator attention between letters. The Text and Filer are set up in a specific but ordinary way. *DeskMate* is then exited, BASIC is called, and *CORMAIL.BAS* is loaded." The program disk contains sample files from Text and Filer to help you along.

To find out more about *CorMail* or to place an order, you can reach Dr. Corwin at Corwin Software, 10066 West Mawman Avenue, Waukegan, IL 60087-2431, (312) 623-4114.

The version currently available runs in interpreted BASIC, with a compiled version planned for the near future. This sounds like a handy program and I will, as I said, let you know more when I see a copy running.

Embedded Printer Codes

I also received a great suggestion from Jerry Stajduhar, in New Hampshire, explaining how to embed printer codes into your *DeskMate* files using the LPDRVR.SYS supplied on your MS-DOS diskette. The process is quite involved but not that difficult. I'll give you complete details next month, but if you're interested in doing some poking around on your own, here are the basics of the process.

As Jerry points out, "the trick to getting control codes into the file is on page 314 of the MS-DOS reference manual." If you don't have this manual, don't worry, I'll give you all the details and explain each step, carefully, next month.

The process is simple; instruct the LPDRVR.SYS software to intercept certain, little-used characters and replace them with strings that contain printer control codes. In order to make this work, you must know how and what codes to use on your brand and model of printer. While many printers use the same codes, many do not. Refer to your printer manual or contact the manufacturer if you need more information about your printer.

The first step is to make sure that DEVICE=LPDRVR.SYS is in your CONFIG.SYS file. Next, you add a PRINT CHR.DAT /P command to your AUTOEXEC.BAT file. The CHR.DAT file contains the codes to replace the old ASCII codes. You must create this file using either a BASIC program like Jerry sent (see the listing) or using *EDLIN*, the DOS line editor supplied on your MS-DOS diskette.

When you use Jerry's program, make sure you name the file (answer the first prompt) with the same name as appears in your AUTOEXEC.BAT file. If you use *EDLIN*, open the file with the name you use in the AUTOEXEC.BAT file. In this example the name is CHR.DAT. I will explain both in more detail next month.

I've only very quickly explained this process and I apologize for not having the room to cover everything every month. In addition, Jerry's letter got here just in time for me to tell you about it, but not in time to try out all the possibilities and put together some listings and examples. In the meantime, look over your printer manual, fix up your work sheet amortization table with a new equation and look over the program Mr. Stajduhar sent in.

In closing, let me thank everyone who wrote in with solutions and suggestions. I know everyone who benefits from these solutions and suggestions thanks you also. It's this type of participation that makes it possible for all of us to make the most of our computers and *DeskMate*. If you have any comments, suggestions, or complaints about this month's material or *DeskMate* in general, just write to me or contact me on Delphi. □

The listing:

```
10 CLS
20 INPUT"ENTER FILE NAME";A$
25 OPEN "O",1,A$
30 Q=1:Q$=""
35 PRINT"TO TERMINATE ENTER A 0"
40 WHILE Q<>0
50 INPUT"ENTER ASCII VALUE TO DUMP TO FILE";Q
55 IF Q=0 THEN 70
60 Q$=Q$+CHR$(Q)
70 WEND
80 PRINT#1,Q$;
90 CLOSE
100 ' BY JERRY STAJDUHAR
110 ' CREATES CHR.DAT FOR LPDRVR.SYS
```

PCM

BASIC's Magnificent Shell Command

By Brian H. Alsop

BASIC has a super-powerful command called SHELL. It can unlock the power of your machine. How would you like a system of menus that can ask questions and decide on the course to follow? SHELL can call BASIC programs, commercial software programs, assembly language programs or batch files. Would you like to access DOS commands such as SORT and DIR directly from BASIC? SHELL can do this, too. While others are struggling with batch files, you can have a command system that runs circles around anything batch files can do. SHELL can do all this and more — and it's easy.

Background

In the beginning there was a computer. It needed an operating system (OS). The purpose of the system was to permit simple operations without having to punch in hexadecimal numbers. Then came BASIC. Its purpose was to create programs to run. In the beginning they were completely separate. The OS couldn't use BASIC commands and BASIC couldn't use OS commands. This was unfortunate. BASIC needed at least some of the OS commands. At first, substitute commands such as FILES for DIR were created, then it became obvious more was needed. There came a bright idea: Why not let BASIC access any operating system command? Why not also let it run other non-BASIC programs and return to BASIC where it left off? The SHELL command was invented.

This story is fictional but describes a history that could have led to the current situation with MS-DOS and BASIC. Anyone who has used a system

Brian Alsop is an engineer working on introducing personal computers into the engineering work environment. He is a pilot and holds an amateur radio operator's license. Brian can be contacted at 113 Boone Rd., Trafford, PA 15085; (412) 373-3363.

where all OS commands are integral to BASIC (like the Radio Shack Color Computer) can't imagine why they would ever be separate. Whatever the reason, it's a fact. DOS commands are not a part of BASIC as implemented on Tandy 1000, 2000 or 3000 computers. They aren't part of BASIC on IBM PCs either. However, it turns out that the SHELL command plus the operating system utilities provided with the OS more than make up for this oversight.

Shell Command Basics

The syntax for the SHELL command is SHELL S$. Listing 1 shows a simple BASIC program invoking the SHELL command. The S$ string can be any DOS command, a batch file or another (non-BASIC) program. When a SHELL is encountered, the area of memory BASIC occupies is protected (including all variables and data), the command specified in the S$ string is executed and then returns to BASIC. The BASIC display screen may be altered.

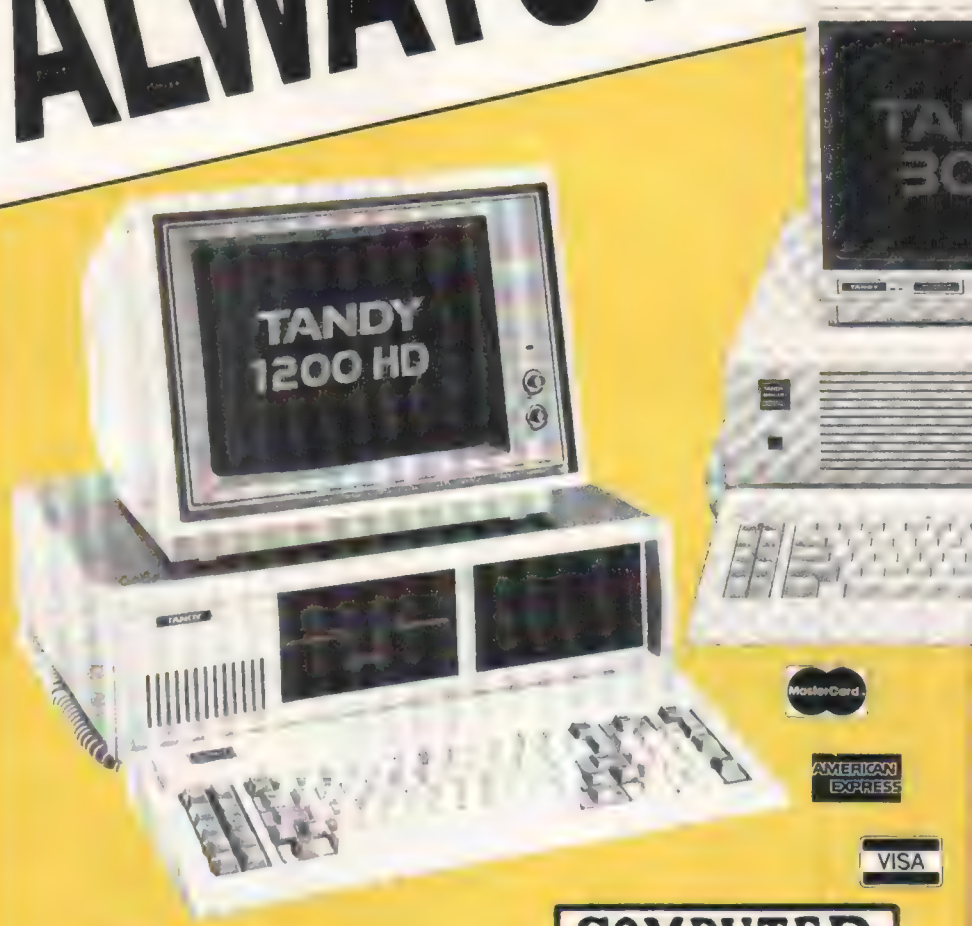
Table 1 gives a listing of some useful SHELL strings. An example of a more complicated string might be SORT <FILE1.ASC >FILE2.ASC. This second string takes FILE1.ASC, sorts it in alphabetical order and places the contents in FILE2.ASC. This could be quite useful. If FILE1.ASC were previously created and closed in the BASIC run, it could be sorted using the DOS assembly language sorting routine and then return the sorted file to BASIC. Since the SORT command also allows fields to be sorted, one could sort on selected fields. This gives BASIC the power of DOS assembly language commands without the mess of using USR functions. Listing 2 illustrates this application of SHELL. The output of the program is also given.

Listing 3 shows how to obtain a sorted directory listing. It combines two SHELL commands in sequence. The program first issues a SHELL"DIR >XYZ." command. BASIC then puts itself on hold and DOS takes over. The directory is listed into file XYZ. The contents of XYZ look exactly like the screen listing when a DIR is issued. This means there are a few lines of header information to skip over before getting to the filenames. After the DOS command is completed, BASIC is reentered and continues where it left off.

The program then issues a SORT <XYZ. >DR. command. This sorts the

TANDY® COMPUTERS
WITH THE MANUFACTURERS WARRANTY

ALWAYS AT "S



MasterCard
AMERICAN EXPRESS
VISA

- DISK DRIVES
- PRINTERS
- MODEMS
- SOFTWARE
- ALL RADIO SHACK PRODUCTS
- CHECK OUR SALE PRICES -- CALL

COMPUTER CENTER

DFW COMPUTER CENTER

Customer Service 817-481-SALE (9 am–5 pm CT)

ALE" PRICES

Do You Want A Real Warranty. . .
or will you settle for a vague promise?

Since 1977, we have operated a Radio Shack Authorized Sales Center (ASC), in the city of Grapevine, Texas. This will be significant for you only if (a) you want a warranty which you can exercise easily, if necessary and (b) the reassurance of our long term stability and business ethics.

When you buy a Tandy/Radio Shack product from us the Tandy/Radio Shack Warranty will accompany it and the warranty and service will be available to you, anywhere in the U.S.

We will not install any foreign parts which might, and probably will, have an adverse affect upon your warranty. We will assist you in obtaining local support, should you experience difficulty, and will make an offer to buy-our-product-back* (hardware) if it dissatisfies you, within 30 days.
"We ship fast," has always been our motto and if anyone is offering you a lower price, we suggest you ask about the foreign parts (and warranty) before closing the deal. We will "NOT meet-or-beat" a lower price so you can be sure you have our best quote the first time, and our toll-free phone lines (outside Texas) make the quote easy for you to get, from 9 a.m.–5 p.m. central time, Monday thru Friday. (*For a small handling charge)

-------TOLL FREE 1-800-433-SALE

326 Main St. Grapevine, Texas 76051

TEXAS BUYERS ADD 5⅛% SALES TAX

XYZ file alphabetically into file DR. Because the header information and other directory information are preceded by spaces, the sorting process puts them all at the top of the DR file. BASIC is then reentered. It opens file DR and begins reading a record at a time. It skips over the header information, retrieves the first 12 characters of each line and writes this filename extension data to the screen.

Up to this point, life was easy — just issue the magic SHELL command. However, if you were to run one of these programs, you would see that the DOS command sometimes writes to the screen with its own comments. These commands either disappear with no effect on the existing BASIC screen or cause the screen to scroll. If this is a problem in a particular application, there is a simple solution. Save the screen to memory before executing the SHELL command and restore it on returning.

Listing 4 shows the augmented subroutine for SHELLing. The BSAVE command saves the screen in file SCREEN.BIN and the BLOAD command calls it back. The BSAVE command needs to know where to start saving from, how much to save and the filename. BSAVE begins saving at a memory location defined by the last DEF SEG command. If none is supplied, it assumes the segment that BASIC occupies.

In a Tandy 1000, the text and graphics screens are located at &HB8000. Therefore, one must position-define the starting point at memory location &HB8000. The DEF SEG = &HB800 performs this function. (Note: There seems to be one zero left off. It is not a mistake. BASIC inserts the extra zero when performing the segment repositioning.) The filename chosen here is SCREEN.BIN. Other Tandy 1000 compatible machines may locate monochrome screens elsewhere in memory. However, their color screens usually start at &HB8000.

How much memory to save depends on the screen mode. The appropriate values for 'X' in the augmented SHELL subroutine are &H1000, &H4000, &H4000, &H4000, &H4000, &H8000 and &H8000 for screen modes zero through six, respectively.

The next DEF SEG command returns to the core memory area BASIC is using after the BSAVE. The BLOAD command needs to know where to load the file in memory and the filename. Again a

Table 1
Useful DOS Commands for SHELL

Command	Function	Sample S$
CD\	Change Directory	"CD\UTILITY"
COPY	Copy a file	"COPY FILE1.BAS B:FILE2.BAS"
DATE	Set Date	"DATE"
DIR	Obtain Directory List	"DIR A:"
DISKCOPY	Copy a Whole Disk	"DISKCOPY A:"
ERASE	Erase a File	"ERASE FILE1.ASC"
FORMAT	Format a Disk	"FORMAT B:"
PRINT	Spool Print a File	"PRINT FILE1.ASC"
RENAME	Rename a File	"RENAME FILE1.ASC FILE2.ASC"
SHIPTRAK	Store HD Head	"SHIPTRAK"
SORT	Sort a File	"SORT <FILE1.ASC >FILE2.ASC"
TYPE	Type File on Screen	"TYPE FILE1.ASC"
TIME	Set Time	"TIME"

DEF SEG =&HB800 is used to define the starting memory location. The previously saved screen is then loaded. A DEF SEG command returns to the BASIC program memory area. It's a good idea to position the cursor where you want it when exiting the SHELL subroutine.

These examples are building blocks to more complicated applications. I use the SHELL command to build complex menus calling DOS commands, BASIC programs, compiled BASIC programs, assembly language programs, FORTRAN programs and batch files. SHELL has been useful in creating a hard disk database for easy location of files. I've even used SHELL to automatically back up and restore disk files to and from mainframe computers using terminal emulation file transfer commands. With SHELL, one is limited only by imagination.

Two more complicated SHELL examples follow. The first is a menu program that accesses all types of BASIC and non-BASIC programs. The second example is the hard disk file database program. Study them, try them, then go off and create your own applications. You'll find BASIC's SHELL command is simpler and more powerful than using batch files or some other lower level language.

Menu Program

A general menu program should permit a high degree of user interaction. It should be able to run programs and batch file sequences of all types. Finally, it should provide for return to itself on the completion of the selected application. With BASIC and the SHELL command this is all possible. The example also illustrates how to get around the problem of executing a compiled BASIC program from within a BASIC menu. Normally, one can't call compiled BASIC from BASIC because they use the same memory area.

Listing 5 shows a sample menu program. It has six choices. From the menu one can invoke spooler printing of a file, run an interpreted BASIC program, run a compiled BASIC program, generate a sorted directory listing using DOS commands, run a batch file and run an assembly language program.

It is assumed that all programs exist on the current drive and directory. Use path suffixes to access programs from other drives and directories.

Lines 50-190 define the program setup and menu. Menu creation is straightforward and involves a few PRINTs and one INPUT command. Once a choice is made, it is first tested to see if the program should end. An 'E' or 'e' entry signifies the end of the program. A SYSTEM command exits the program. Next the value of the input variable X$ is determined. An ON .... GOSUB multiway branch is used to select the appropriate subroutine. On return from the subroutine, the screen is cleared and the menu rewritten.

The first choice presented is sending a file to the *DOS Print* program to be printed in the background while program execution continues. The program executes the subroutine in lines 210-250 to do this. First the program prompts for the filename to spool. One can specify a full path and filename. Then the program constructs the DOS PRINT command by linking the filename to *Print* and puts this string into S$. Next a SHELL S$ is executed. The operating system takes over and loads *Print* (if it is not already resident) and spools the file to the printer. The *Print* program reports on what files it has in

Listing 1: Simple SHELL Example

```
10 'This program copies a file named X.BAT to Y.BAT
20 S$="COPY X.BAT Y.BAT"
30 GOSUB 200
40 END
200 SHELL S$
210 RETURN
```

Listing 2: File Sorting in BASIC using DOS SORT

```
10 ' First a file with non-ordered strings is created
20 OPEN "O", #1, "FILE1.ASC"
30 PRINT #1, "XYQ"
40 PRINT #1, "ABC"
50 PRINT #1, "DEF"
60 CLOSE
70 S$="SORT <FILE1.ASC >FILE2.ASC"
80 GOSUB 200
90 OPEN "I",#1, "FILE2.ASC"
100 IF EOF(1) THEN END
110 LINE INPUT#1, A$
120 PRINT A$,
130 GOTO 100
140 END
200 SHELL S$
210 RETURN
```

```
OUTPUT FROM LISTING 2
ABC         DEF         XYQ
```

its queue and releases control. At this point BASIC is reentered and the menu rewritten. You can continue doing something else while the document is printing.

The second choice is running an interpreted BASIC program. The subroutine in lines 260-270 do this. The process is simple. Just execute a RUN-"BASPROG2". That command loads the program and executes it. To get back from *BASPROG2*, one must insert a line RUN"MENU" in *BASPROG2* to return to this *Menu* program. This line replaces the normal END or SYSTEM line in *BASPROG2*.

BASPROG2 can do anything including specifying more menu choices. I use this extensively to permit others in the household to get to their programs without having to remember filenames. They just select menu choices. If the RUN"BASPROG2" command is preceded by a SHELL command redefining the directory, multiple users could automatically be sent to their directory before accessing any files. Returning to the main menu in this case would require a SHELL "cd\" line followed by a RUN "MENU" line — assuming *Menu* was in the root directory.

The third choice is running a compiled BASIC program. The subroutine in lines 280-330 shows how. There are many of these programs on the market today. Public domain bulletin boards are loaded with them. Even some commercial software programs are written in compiled BASIC. These programs typically have an extension of EXE. The difficulty in accessing them via interpreted BASIC is that they both need the same work space. This memory conflict can easily be resolved by exiting BASIC, loading the program and reloading the interpreted BASIC menu program on completion. The easiest way to accomplish this is to create a batch file. In this case, the name is BAT1.BAT. The batch file is then SHELLed. The first line in the batch file is the compiled BASIC program to be run. The second line is BASICA MENU. If some communication is needed between the programs, do it via files.

The fourth choice is obtaining a sorted directory. This subroutine is in lines 340-490. It is identical to Listing 2. The only enhancement added is the addition of a prompt to return to the menu.

Listing 3: Sorted Directory Program

```
10 ' THIS PROGRAM OBTAINS A SORTED DIRECTORY LISTING
20 GOSUB 340
30 END
340 ' THIS SUBROUTINE OBTAINS A SORTED DIRECTORY
350 SHELL"DIR>XYZ."
360 SHELL"SORT <XYZ. >DR."
370 CLS
380 OPEN "I",#1,"DR."
390 FOR I=1 TO 1000
400 IF EOF(1) THEN 490
410 INPUT #1, A$
420 IF I=4 THEN PRINT A$
430 IF I<6 THEN 480
440 IF LEFT$(A$,1)=" " THEN 480
450 IF LEFT$(A$,1)="." THEN 480
460 PRINT LEFT$(A$,12)+"  ";: CT=CT+1
470 IF CT>4 THEN PRINT:CT=0
480 NEXT I
490 RETURN
```

Listing 4: Augmented SHELL Subroutine

```
200 DEF SEG = &HB800
210 X=&H1000 :'TEXT SCREEN
210 BSAVE "SCREEN.BIN", 0, X
220 DEF SEG
230 SHELL S$
240 DEF SEG = &HB800
250 BLOAD "SCREEN.BIN",0
260 DEF SEG
270 RETURN
```

The fifth choice is running a batch file. Lines 500-570 contain the commands. This example allows resetting of the date and time from within BASIC. The only difference between this example and the compiled BASIC choice is that BASIC does not have to be reloaded to return to it. In this example, the *Date* and *Time DOS* programs prompt for the needed input. This is potentially a case where one may want to save the BASIC screen before executing the SHELL.

The sixth choice presented is running an EXE or COM program from BASIC. The subroutine in lines 580-640 show how to do this. This example SHELLs the *DOS Debug* program. When *Debug* is exited, the computer returns to the BASIC menu program. I use this type of SHELL extensively to create interactive pre-processors and post-processors for existing FORTRAN programs. All the input is interactively generated in a BASIC program and written to a file. The FORTRAN program reads the input file, runs and stores its output in a file. The BASIC program reads the file, displays it, plots it or whatever. Changes to the input can be made and the FORTRAN job resubmitted. It's like having the best of both worlds.

Lines 650-680 contain the error subroutine. Only one type of error is specifically flagged. If a file is not found, an error message saying so is printed on Line 25. After a delay, the program is resumed at Line 70. Other errors generate no message and resume at Line 70.

Hard Disk File Database Program

One of the problems with a computer is keeping track of the programs. If you have hard disk, this task requires super-human efforts. I needed a program to keep track of every file's location and allow me to find it — given a filename or partial filename. Also, it should be able to update itself, when requested, to reflect changes. Printed output should be available when requested.

After some thought, it became obvious that SHELL was the way to do it. Listing 6 shows the result. The process is simple. Obtain sorted directory listing files for all directories, then

read these files and search for selected filenames or filename substrings. From our previous examples, we already know how to create the sorted directories. All one has to do is write a small program to permit reading and searching of the files. It turned out the program could also find all directory names, all files written on a particular date and all files of a particular extension. Just tell the program the appropriate substring on which to search. The program functions as follows.

Lines 100-170 read the directory names from the DATA statements. This list of directory names is terminated with a STOPIT as the last entry. These directory names are entered into array FF$ starting with Array Index 2. Array Index 1 is reserved for the root directory. One can also handle subdirectories by entering the path and subdirectory name (e.g., wordstar\direct2). With the exception of subdirectories, no backslash characters are entered (see lines 1000-1010).

Next, these directories are put in alphabetical order by a call to the subroutine at Line 1020. Lines 200-230 construct GF$ strings. GF$ contains the actual path statement and filename for the to-be-created DR sorted directory files. Lines 250-260 similarly set up GF$ for the root directory.

Lines 270-570 contain the menu and option-selection coding. First the program asks in Line 320 if the directory files are to be updated. If so, the program goes to the update subroutine starting at Line 810. This subroutine is very similar to that discussed before. The only real difference is the process is in a loop to generate separate sorted directory files for all directories. This requires a CD command to switch to the appropriate directory before generating the file.

Line 960 returns the directory to the root before continuing. Assuming the sorted directory files DR exist, the program then asks if one is interested in searching for files in all directories or just one. If an 'E' is entered, the program ends. If a single directory is chosen, one can update just that directory, print the sorted directory, or search for a file or substring.

If all is chosen, the program allows searching for a particular file or substring only. Lines 580-770 do the searching of all DR files or the particular directory's DR file. The BASIC INSTR command is used to look for the specified substring or filename. The screen output is paginated and all matches found are printed along with their file size, date and directory. The program returns to the menu after completing the search.

That's it. The key to performing this was the ability to use DOS commands within BASIC.

Other Applications

To obtain the most from the SHELL command, become familiar with all the programs supplied with MS-DOS. Barden and others have written articles in PCM explaining how to use these programs. This article should provide enough information to integrate other commercial programs as well. For starters, try writing a hard disk backup program to back up multiple diskettes of data and to prompt when to change diskettes. □

Listing 5: Menu Program

```
10 ' MENU PROGRAM
20 ' ILLUSTRATING HOW TO USE SHELL AND R
UN COMMANDS TO
30 ' ACCESS DOS COMMANDS, BASIC AND NON-
BASIC PROGRAMS
40 ' VIA MENU CONTROL
50 ON ERROR GOTO 650 :'ERROR HANDLING RO
UTINE
60 'SET UP MENU
70 KEY OFF:COLOR 0,14,0 :CLS
80 LOCATE 3,33:PRINT "MENU CHOICES"
90 LOCATE 5,27:PRINT "1. SEND FILE TO PR
INT QUEUE"
100 LOCATE 7,27:PRINT "2. RUN BASIC PROG
RAM  "
110 LOCATE 9,27:PRINT "3. RUN COMPLIED B
ASIC PROGRAM"
120 LOCATE 11,27:PRINT"4. LIST SORTED DI
RECTORY"
130 LOCATE 13,27:PRINT"5. RUN BATCH FILE
140 LOCATE 15,27:PRINT"6. RUN EXE FILE
"
150 LOCATE 18,35:PRINT"E. END"
160 LOCATE 21,27:INPUT"MAKE CHOICE: ",X$
170 IF X$="E" OR X$="e" THEN SYSTEM
180 ON VAL(X$) GOSUB 210,260,280,340,500
,580
190 GOTO 70
200 END
210 ' THIS SUBROUTINE SENDS A FILE TO TH
E PRINT QUEUE.  THE DOS PRINT SPOOOLER
      PROGRAM IS USED.
220 LOCATE 25,1:INPUT"INPUT FILENAME TO
PRINT ";X$
230 S$="PRINT "+X$
240 SHELL S$
250 RETURN
260 ' THIS SUBROUTINE RUNS BASIC PROGRAM
 #2.  TO RETURN TO THIS MENU THE PROGRAM
      MUST CONTAIN A RUN "MENU".
270  RUN"BASPROG2"
280 ' THIS SUBROUTINE RUNS A COMPILED BA
SIC PROGRAM.  HERE A BATCH FILE MUST BE
      EXECUTED.  THE BATCH FILE BAT1 CON
TAINS ARE:
290 '                     COMPBAS1.EXE
300 '                     BASICA MENU
310 '  THE BATCH FILE FIRST EXECUTES THE
 COMPILED BASIC PROGRAM COMPBAS1 AND
       THEN RERUNS THIS MENU PROGRAM
320 SHELL"BAT1.BAT"
330 SYSTEM
340 ' THIS SUBROUTINE OBTAINS A SORTED D
IRECTORY
350 SHELL"DIR>FILE1.ASC"
360 SHELL"SORT <FILE1.ASC >FILE2.ASC"
370 CLS
380 OPEN "I",#1,"FILE2.ASC"
390 FOR I=1 TO 1000
400 IF EOF(1) THEN 490
410 INPUT #1, A$
420 IF I=4 THEN PRINT A$
430 IF I<6 THEN 480
440 IF LEFT$(A$,1)=" " THEN 480
450 IF LEFT$(A$,1)="." THEN 480
460 PRINT LEFT$(A$,12)+"  ";: CT=CT+1
470 IF CT>4 THEN PRINT:CT=0
480 NEXT I
490 PRINT:LOCATE 25,1:INPUT"HIT ENTER TO
 CONTINUE",X$:RETURN
500 'THIS SUBROUTINE RUNS A BATCH FILE W
```

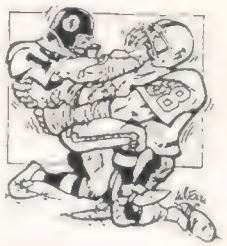
```
HICH UPDATES DATE AND TIME
51Ø 'THE BATCH FILE NAME IS DTETIM.BAT.
IT CONTAINS:
52Ø '                          DATE
53Ø '                          TIME
54Ø CLS
55Ø SHELL"DTETIM.BAT"
56Ø LOCATE 25,1:INPUT"HIT ENTER TO RETUR
N ",X$
57Ø RETURN
58Ø ' THIS SUBROUTINE RUNS AN EXE OR COM
 FILE.  EXE FILES CAN SOMETIMES BE
     COMPILED BASIC PROGRAMS.  IF STRAN
GE THINGS HAPPEN WHEN YOU RETURN (E.G.
59Ø '  DOUBLE CURSORS OR WORSE) YOUR EXE
 FILE WAS MOST LIKELY A COMPILED BASIC
6ØØ ' PROGRAM.  TREAT IT LIKE THE COMPIL
ED BASIC EXAMPLE ABOVE.
61Ø CLS
62Ø SHELL"DEBUG.EXE "
63Ø CLS:LOCATE 25,1:INPUT"HIT ENTER TO C
ONTINUE ",X$
64Ø RETURN
65Ø 'ERROR ROUTINE
66Ø IF ERR=53 THEN BEEP:LOCATE 25,1:PRIN
T"FILE NOT FOUND"
67Ø FOR I=1 TO 2ØØØ:NEXT
68Ø RESUME 7Ø
```

Listing 6: Hard Disk File Inventory Program

```
1Ø T1$="                              FINDER P
ROGRAM"
2Ø T2$="COPYRIGHT 1986- B.H. ALSOP, 113
BOONE RD., TRAFFORD, PA. 15Ø85"
3Ø ' THIS PROGRAM FINDS A FILE OR PARTIA
L FILE NAME, EXTENSION OR SUBSTRING IN
     A SPECIFIED DIRECTORY OR IN ALL DIR
ECTORIES.
4Ø ' SORTED DIRECTORIES OF FILES CAN BE
PRINTED.
5Ø ' THIS PROGRAM ILLUSTRATES THE USE OF
 SHELL COMMANDS TO EXECUTE DOS COMMANDS
     FROM WITHIN BASIC.  THE DOS CD\, DI
R, SORT AND FILE REDIRECTION COMMANDS
     ARE USED.
6Ø ' UP TO 33 DIRECTORIES CAN BE HANDLED
 WITHOUT REFORMATING THE MENU.
7Ø COLOR 14,1,1:KEY OFF:CLS
8Ø LOCATE 1Ø,1Ø:PRINT T1$:LOCATE 14,1Ø:P
RINT T2$:FOR I=1 TO 3ØØØ:NEXT
9Ø N=34
1ØØ 'DIMENSION AND READ DIRECTORY NAMES
FF$(I)
11Ø 'STOP READING WHEN "STOPIT" IS READ
12Ø 'PROGRAM CREATES A SORTED FILE OF DI
RECTORY NAMES IN A FILE NAMED DR
13Ø DIM FF$(N+1),GF$(N+1)
14Ø FOR I=2 TO N
15Ø READ FF$(I)
16Ø IF FF$(I)="STOPIT" OR FF$(I)="stopit
" GOTO 19Ø
17Ø NEXT I
18Ø GOSUB 1Ø2Ø
19Ø N=I-1
2ØØ 'ADD PATH NAMES TO FF$ AND CONSTRUCT
 STRINGS LIKE \BASIC\DR IN GF$(I)
21Ø FOR I=2 TO N
22Ø GF$(I)="\"+FF$(I)+"\DR."
23Ø NEXT I
24Ø 'SET UP ROOT DIRECTORY AS FIRST ENTR
Y IN FF$(I) AND GF$(I)
25Ø GF$(1)="DR."
26Ø FF$(1)="ROOT"
27Ø 'SET UP SCREEN MENU.  MORE THAN 17 D
IRECTORIES PRODUCES 2 COLUMN MENU
28Ø COLOR 14,1,1:CLS
29Ø IN=16
3ØØ  LOCATE 1,28:PRINT"SORTED DIRECTORY
PROGRAM"
31Ø 'UPDATES OF DIRECTORY CONTENTS ARE M
ADE BY CHOOSING Y
32Ø LOCATE 3,26:INPUT "SORT ALL DIRECTOR
IES(Y/N)";UD1$
33Ø IF UD1$="Y" OR UD1$="y" THEN 34Ø ELS
E 35Ø
34Ø FOR M=1 TO N:LOCATE 15,25:PRINT"UPDA
TING DIRECTORY "; FF$(M);:LOCATE 25,1:GO
SUB 81Ø:NEXT M
35Ø COLOR 14,1,1:CLS
36Ø LOCATE 1,28:PRINT"SORTED DIRECTORY P
ROGRAM"
37Ø LOCATE 3,1
38Ø FOR I=1 TO N
39Ø Y=2+I
4ØØ IF I>17 THEN DN=32 ELSE DN=Ø
41Ø IF I>17 THEN Y=Y-17
42Ø LOCATE Y,IN+DN:PRINT USING"##";I;:PR
INT". ";FF$(I);
43Ø NEXT I
44Ø 'PROMPT FOR DIRECTORY, E TO EXIT OR
ALL FOR ALL DIRECTORIES
45Ø LOCATE 21,5:INPUT; "CHOOSE ONE (E TO
 EXIT, ALL FOR ALL DIRECTORIES)";X$
46Ø IF X$="E" OR X$="e" THEN SYSTEM
47Ø IF X$="ALL" OR X$="all" THEN GOTO 56
Ø
48Ø IF VAL(X$)>N OR VAL(X$)<1 THEN 45Ø
49Ø LOCATE 22,15:INPUT;"PRINT (Y/N)";LP$
:'PRINT OPTION
5ØØ IF X$="E" OR X$="e" THEN SYSTEM :'EX
IT
51Ø IF X$="ALL" OR X$="all" THEN GOTO 56
Ø
52Ø M=VAL(X$):IF M>N THEN GOTO 45Ø
53Ø IF UD1$="Y" OR UD1$="y" THEN GOTO 56
Ø ELSE LOCATE 23,15:INPUT;"UPDATE SORTED
 LISTING (Y/N)";UD$
54Ø IF UD$="Y" OR UD$="y" THEN GOSUB 81Ø
55Ø 'PROMPT FOR FILE NAME OR PHRASE PP$
TO SEARCH FOR
56Ø LOCATE 24,1:INPUT;"SEARCH FOR PHRASE
(Y/N)";P$
57Ø IF P$="Y" OR P$="y" THEN LOCATE 24,1
:PRINT STRING$(4Ø," ");:LOCATE 24,1:INPU
T"INPUT PHRASE(ALL CAPS)";PP$
58Ø 'SEARCH DR FILES FOR SUBSTRING PP$
59Ø CLS
6ØØ IF X$="ALL" OR X$="all" THEN GOTO 61
Ø ELSE GOTO 62Ø
61Ø FOR M=1 TO N
62Ø OPEN "I", #1, GF$(M)
63Ø FOR I=1 TO 1ØØØ
64Ø LINE INPUT #1, A$
65Ø IF I<3 THEN GOTO 72Ø
66Ø IF P$="Y" OR P$="y" THEN V=INSTR(A$,
PP$):IF V>Ø THEN 67Ø ELSE 7ØØ
67Ø PRINT "        "; A$,:IF I>5 THEN PRI
NT FF$(M) ELSE PRINT
68Ø K=K+1
69Ø IF LP$="Y" OR LP$= "y" THEN LPRINT A
$,FF$(M):K=Ø
7ØØ IF EOF(1) THEN GOTO 73Ø
71Ø IF K>2Ø THEN LOCATE 25,1:BEEP:INPUT;
```

Pigskin Predictions! Pro Handicapper

Tired of wrestling with Sunday point spreads? Let your Tandy 1000/2000 do it for you! Pigskin Predictions, the best-selling NFL handicapper, takes the hassle out of rating National Football League games. Forget about obscure, meaningless statistics. Just spend a few minutes typing in each week's scores and let our point spread software go to work. Here's what Pigskin Predictions has to offer:



- Predicts point spreads for all games--for the current week and the remainder of the season.
- Calculates projected won-lost records for all weeks.
- Menu-driven selection of schedules, ratings, division races, predictions or results by team or week. Seven different reports to screen or printer!
- Maintains home field advantage and power ratings for all NFL teams
- 1986 Schedule data file included free. Yearly updates available.

Pigskin Predictions is only $39.95 on disk. Versions also available for Tandy Color Computer and Commodore 64/128. Get your copy now and be ready for the season!

Use your Tandy 1000, 1200, 2000 or Model 100/200 to improve your performance at the track! Separate programs for **Thoroughbreds, Harness** horses and **Greyhounds** rank the horses or dogs in each race quickly and easily, even if you've never handicapped before!

All the information you need is readily available in the thoroughbred *Racing Form*, harness or dog track program. We even provide a chart showing you exactly where to get the information you need! Our software puts the savvy of a veteran handicapper at your fingertips. Our complete instructions and wagering guide tell you how to bet, which races to bet and which ones to avoid—one of the real secrets of winning at the track!

Thoroughbred factors include speed, distance, past performance, weight, class, jockey's record, beaten favorite and post position. **Harness factors** include speed, post position, driver's record, breaking tendencies, class, parked-out signs and beaten favorite. **Greyhound factors** include speed, past performance, maneuvering ability, favorite box, class, kennel record, beaten favorite and breaking ability.

Thoroughbred, Harness and **Greyhound** programs are sold separately. **Tandy 1000, 1200** and **2000** disk versions are $49.95 each, any two for $74.95, all three $99.95. **Thoroughbred and Harness** only available for Model 100/200 on tape at $49.95 each, both for $79.95.

MasterCard

Federal Hill Software
8134 Scotts Level Rd.
Baltimore, MD 21208

VISA®

Orders 800-628-2828 Ext. 850
Information 301-521-4886


```
"HIT ENTER TO CONTINUE";R$:K=Ø:CLS:LOCAT
E 2,1
72Ø NEXT I
73Ø IF X$="ALL" OR X$="all" THEN CLOSE:G
OTO 75Ø
74Ø LOCATE 25,1:BEEP:INPUT;"HIT ENTER TO
 CONTINUE",R$:CLOSE:K=Ø:LOCATE 1,1:CLS
75Ø IF X$="ALL" OR X$="all" THEN NEXT M
76Ø IF X$="ALL" OR X$="all" THEN LOCATE
25,1:BEEP:INPUT"HIT ENTER TO CONTINUE";R
$
77Ø GOTO 35Ø
78Ø END
79Ø 'UPDATE SORTED DIRECTORY
8ØØ 'CREATE STRING S$ WHICH CONTAINS DOS
 COMMAND TO CHANGE DIRECTORIES
81Ø S$="CD\"+FF$(M)
82Ø IF M=1 THEN S$="CD\"
83Ø 'EXECUTE DOS CHANGE DIRECTORY COMMAN
D VIA SHELL WITHIN BASIC
84Ø SHELL S$
85Ø 'CREATE STRING S$ TO DO A DIR AND RE
DIRECT THE CONTENTS FOR FILE XYZ
86Ø S$="DIR >XYZ."
87Ø 'EXECUTE DIRECTORY COMMAND
88Ø SHELL S$
89Ø 'CREATE STING TO SORT DIRECTORY FILE
 XYZ ALPHABETICALLY INTO FILE DR
9ØØ S$="SORT <XYZ. >DR."
91Ø 'DO SORTING
92Ø SHELL S$
93Ø 'KILL INTERMEDIATE FILE XYZ
94Ø KILL"XYZ."
95Ø 'RETURN TO ROOT DIRECTORY
96Ø SHELL "CD\"
97Ø UD$=""
98Ø RETURN
99Ø 'DATA ENTRIES CONTAIN THE LIST OF DI
RECTORY NAMES
1ØØØ DATA BASIC,BCOMP2,CIRCLEW,COCOUTIL,
DESKMATE,FINANCE,FLY,FORT,LORIE,QUBIE,RT
TY,UTILITY,WP
1Ø1Ø DATA STOPIT
1Ø2Ø 'SORT DIRECTORY NAMES
1Ø3Ø FOR I=2 TO N-1
1Ø4Ø FOR J=I+1 TO N
1Ø5Ø IF FF$(I)< FF$(J) THEN 1Ø7Ø
1Ø6Ø T$=FF$(J):FF$(J)=FF$(I):FF$(I)=T$
1Ø7Ø NEXT J
1Ø8Ø NEXT I
1Ø9Ø RETURN
```

PCM

This $5 hardware project brings new life to old joysticks

Joystick Reincarnation

By Brian H. Alsop

Older joysticks have only one button. Many software programs appearing on the market today require two buttons per joystick. BASIC can also access two buttons. What do you do if you have older joysticks with only one? Reincarnation is the answer. This article describes how to modify older, single-button joysticks to make them functionally equivalent to newer models. The cost is under $5. These modifications specifically apply to the Tandy 1000 and Radio Shack's 26-3008 joysticks. However, the idea can be applied to other joysticks by substitution of appropriate connectors.

Modification

My joysticks are from early Tandy Color Computer days. They functioned correctly with the Tandy 1000 but didn't have the extra button. I priced the new joysticks and decided instead to modify the old ones. The modification turned out to be quite easy.

The Tandy 1000 button and joystick circuitry are shown in Figure 1. The buttons simply ground a pin of the connector. This means the modification only requires adding a button, one wire and possibly a six-pin DIN plug.

The parts needed are a six-pin DIN plug (Radio Shack #274-020); a normally-open, momentary pushbutton switch (Radio Shack #275-1547); a piece of wire about six inches longer than the joystick cable; and some tape.

Proceed with the modification as follows. Refer to Figure 2 for guidance. The modification is far easier to do than to describe.

1. Loosen the two screws holding joystick case together. Remove the top section.

Brian Alsop is an engineer working on introducing personal computers into the engineering work environment. He is a pilot and holds an amateur radio operator's license.

THE PORTABLE

3.5" MICRO FLOPPY DRIVE
Purple Computing Model D103

$229 with TS-DOS software for Model 100/200, NEC PC 8201A

$259 with LAPDOS software for the IBM PC/XT/AT

$299 with both LAPDOS and TS-DOS

At 28 ounces and ⅓ the size of a Model 100, this disk drive is totally portable. The four size "AA" batteries inside, power the drive for 2+ hours of continuous reading/writing at over 2,000 char./sec. (19,200-baud). Fast random file access to 102,400 bytes means no more waiting for cassettes. The best part is the software—you get your choice of either ***TS-DOS*** for the Model 100/200 & NEC PC-8201, or ***LAPDOS*** for the IBM.

For one price we send you everything you need to start using the disk. Even an A/C adaptor (saves batteries when A/C is handy).

COMPLETELY COMPATIBLE!
The disks, formats, data, and everything else are completely compatible and interchangable with the TANDY portable disk drive.

TS-DOS on 3½" diskette for your portable, lets you save and load files with a single key, or save/load all files at once; append files to create one file up to 64K; display or print files directly from disk; access disk files directly from BASIC using OPEN, PRINT #, INPUT #, ...; occupies only 500 bytes when not in use and requires under 5K to run. TS-DOS is supported by ULTIMATE ROM II.

LAPDOS consists of a 5.5" diskette for your IBM PC/XT/AT, manual, and special cable adaptor which lets you hook our Model D103 drive to your IBM's serial port. LAP-DOS supports file conversion between: T-BASE↔DBASE II and IDEA↔ThinkTank, and can run RAM resident!

A $99 PRINTER MADE FOR TRAVEL

SIZE and WEIGHT are the most important considerations when going portable. This printer is half the size of the Model-100 and weighs less. Now there's no reason not to have a hard copy printout wherever you are.

FEATURES

- 2.2 lbs., size 11" x 4¼" x 1¾".
- Epson MX/80 bit-map. graphics.
- Built-in auto self test function.
- 8½" wide thermal paper.
- Excellent User's Manual.
- 40/80/160 chars./line.
- 512 byte text buffer.
- 40 chars./sec.

RUGGED

Because it's thermal, there are no ribbons, no ink, no pins to wear out or replace, all you need to take with you is paper. Prints over two hours on internal batteries and is designed to print over 300,000 lines without maintenance—that's 25 pages every week for over four years!

DIRECT TO YOU

High volume and no middle man means low price. We buy direct from the Japanese manufacturer in large quantities for our corporate accounts (NBC, PanAm, National Geographic . . .), that's why we can offer it direct to you at this fantastic price, yet still give you great service and support.

SUPPORT

We stock paper (rolls and sheets), cables, and accessories for the printer. A complete accessory kit is only $24 and includes a carrying case, a 100' roll of paper, and the hard-to-find 6VDC @ 1.1Amp. A/C adaptor (saves batteries when A/C is handy).

$99 GRAND TOTAL

You get the printer and a specially made 18" cable for your Model 100/200/600 or NEC PC-8201A/8401, a 20' roll of paper, paper holder, and four "C" size batteries—everything you need to start printing immediately.

(ACTUAL print sample)
80 Chars/Line, 40 CPS!
Enlarged!
underlined Emphasized!
CONDENSED gives you 160 chars. per line!

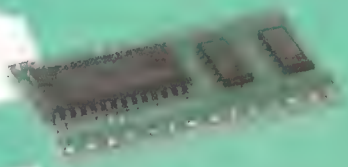
EXPANSIONS!!!

NOT "ALL-THE-SAME"

The memory expansion products sold by us, our dealers, and distributors are made by us—here in the U.S.A. We have been making them since 1983 and have more than 50,000 memory expansion units in service worldwide—our dealers and customers tell us ours are the best. You could spend less or you could spend more, but you won't find a better combination of price, quality, and service than you will from Purple Computing, or one of our dealers.

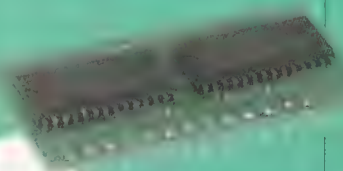
TANDY 600
96K Memory Unit
$249—two for $469

Designed as an alternative to the expansion board that TANDY offers. Functionally identical and easy to install with our complete and detailed documentation—just open the case and plug it in. The basic TANDY 600 machine will accept two of these boards, bringing its internal memory up to 224K (contiguous).



MODEL 100 or NEC PC-8201A
8K Memory Module
$29 ea., 3 for $79, 6 for $149

Easy to install. Open the case and plug-them-in. Each module expands your memory by 8K bytes. Extra low power components mean long battery life. Very low profile means a proper fit in the NEC PC-8201A. Detailed instructions make installation quick and easy. You can expand your Model 100 to 32K and your PC-8201 to 64K (2 banks of 32K each).



TANDY 200
24K Memory Module
$75 ea. or two for $130

WE'LL BEAT ANY *advertised price in this magazine on the 24K Module.*

Simply pop open the hatch with a coin and plug in one or two of these modules. Each module adds a 24K bank of memory to the TANDY 200. It's like getting two more machines. The built in COPY function key copies files between banks instantly. Like our 8K, we build these with the lowest power and most reliable memory chips available.

NEC PC-8201A
128K RAM CARTRIDGE
THE SIDECAR
$259

Plugs into side expansion slot providing four additional banks of 32K each. The case conforms to the shape of the 8201 extending its width only 2¼ inches. An internal battery backs up the memory for over seven years.

STARLET (NEC PC-8401A)
128K RAM DISK CARTRIDGE
THE SIDESTAR
$399

Plugs into the side expansion slot providing a 128K RAM DISK that is formatted and used by the STARLET just like a floppy disk drive—except it's hundreds of times faster, and you don't need a drive, cables or A/C power. Lets you use 64K mode and run larger programs like DBASE II. The expansion connector lets you copy large files (up to 124K) between disk and SIDESTAR using NEC Disk/Video interface. An internal battery backs up memory for over seven years.

When you order, ask for the CP/M utilities and we will load them into your SIDESTAR at no extra charge. (Includes DDT, ED, ASM, LOAD . . .)

THE PURPLE POLICY

Try any of our products for 30 days, satisfy yourself that our service, quality, and prices add up to the best value anywhere—if not completely satisfied, you can return the product for a full refund. Join our (over 20,000) satisfied customers. All prices include free UPS surface shipping, and there are no surcharges for VISA/MC/American Express—even the phone call is free.

IT'S EASY TO ORDER

Send your order with payment to the address below. Or, if you prefer, credit card orders can be handled by phone—VISA, MasterCard, and American Express are welcome. California residents add 6% sales tax. Checks allow 3 weeks to clear. Institutions and Corporations call for purchase order approval.

1-800-732-5012 TOLL FREE
Orders only (8am.-5pm. PST)
In California ***(805) 987-4788***
For orders or customer service

If you would like our latest product catalog or to find a dealer near you, please write or call (805) 987-4788

PURPLE COMPUTING

420 Constitution Ave.
Camarillo, CA 93010
Telex: 888661 (PURPLE)

Canada: Canada Portable Computer, Langley, B.C. (604) 856-8858

Australia: Softech Computer Services, Sydney. (2) 419-8899

Trademarks: TS-DOS, LAPDOS, ULTIMATE ROM, IDEA!, T-base, T-Writer—Traveling Software, Inc., ThinkTank—Living Videotext, Inc., DBASE II—Ashton Tate., CP/M, DDT, ED, ASM, LOAD—Digital Research Inc.

2. Drill a hole to accommodate the pushbutton in the upper left-hand corner on the plastic top section. Drill it about ¾ inches in from the top and left-hand sides.

3. Solder a three-inch piece of wire to one pushbutton terminal. Strip the other end.

4. Solder the long wire (which will go to the plug) to the the other button terminal.

5. Install the button in the top section.

6. Solder the short button wire to the solder lug having the black wire attached. This black ground wire is attached to DIN Plug 3. It may be some other color for other joysticks.

7. Carefully route the long wire through the existing cable exit.

8. Close the case and secure the screws. Make sure the joystick doesn't bind.

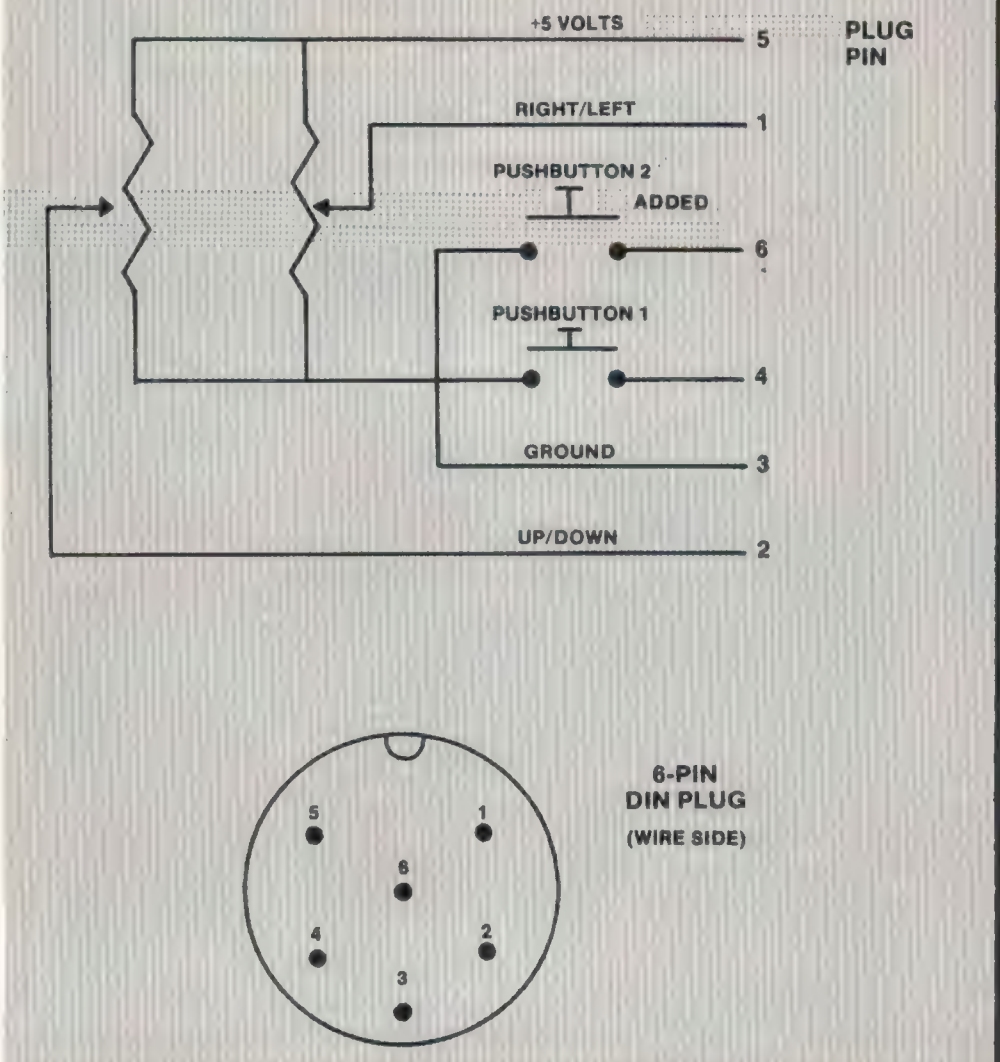
9. Tape the wire along the existing cable every three or four inches.

10. Disassemble the DIN connector by prying up the locking finger and pulling the casing back.

11. If your joystick already has a six-pin DIN connector, solder the added wire to the center pin. Trim off excess wire before soldering. The center pin is labeled Pin 6. Reassemble the connector. Follow steps 12-14 if you have only a five-pin DIN connector.

12. Unsolder all wires. Keep track of which wire went to which pin. In my case, pins 1-5 had wires colored yellow, green, white, black and red, respectively.

Figure 1. Joystick Circuitry



LONESTAR SOFTWARE PROUDLY PRESENTS

LCdraw
LONESTAR

MENU
Seg
Rec
Txt
Cir
Opt
Quit
penup

OPT> [Load Save Print]

CREATE YOUR OWN BANNERS, LOGOS, GRAPHS, CHARTS, AND MORE!

AN INTERACTIVE GRAPHICS EDITOR FOR THE MODEL 100 24K

FEATURES:
*USER FRIENDLY
*ON SCREEN MENU
*CURSOR CONTROL
*SAVEABLE DISPLAY
*PRINT OPTION
*TEXT EDITOR
*FAST MACHINE CODE
*SKETCH MODE
*STRETCHABLE RECTANGLE
*STRETCHABLE SEGMENTS
*CIRCLE OPTIONS
*ANGLE LOCKS
*FILL PATTERNS
*CONSTANT X,Y READOUT
**Coming soon for the MODEL 200!

$49.95

To order call: 1-800-835-2246 ext.298
In Kansas 1-800-362-2421 ext.298
VISA & MASTERCARDS WELCOME

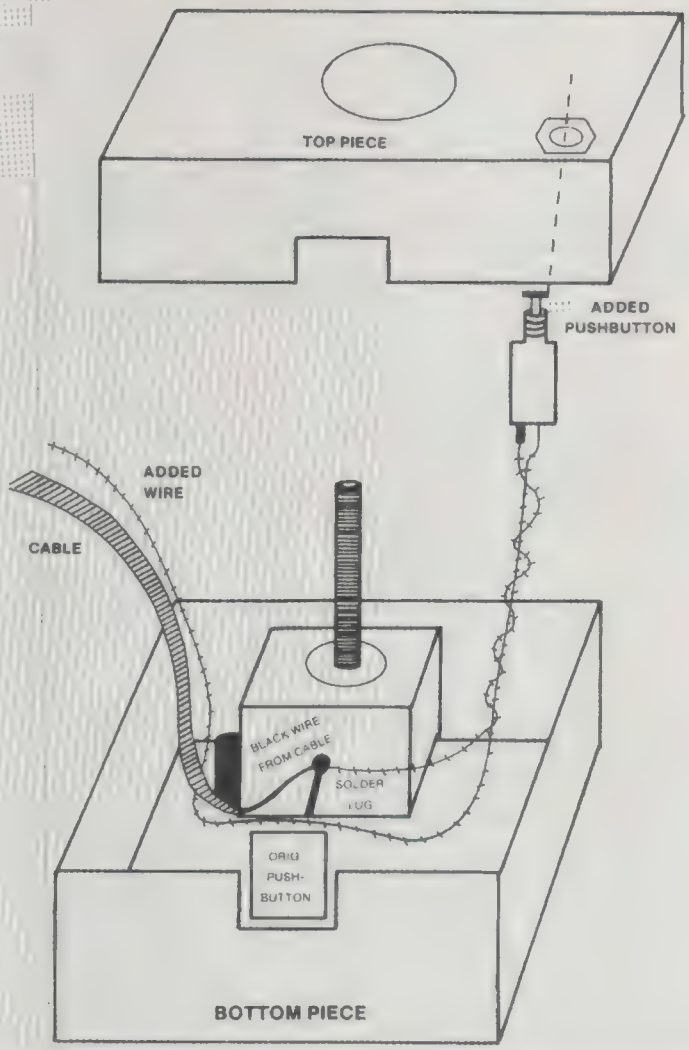
LONESTAR SOFTWARE
& IC DESIGN CO. HOUSTON, TEXAS

13. Remove the old plastic casing from the cable. Slide on the new casing and route the extra wire through it.

14. Solder the extra wire to the central pin (Pin 6). Then, solder on the remaining wires to the correct pins. Reassemble the connector.

The correct button operation can be tested with an ohmmeter or the program shown in Listing 1. If you use an ohmmeter, depress the new switch and check for a short circuit between pins 3 and 6. Open the switch and see that an open circuit condition exists. If you use Listing 1, insert the joystick into the left or right port and run the BASIC program. Depress the added button. The status for button two should change from "off" to "on." If it does not, look and see if any of the other joystick readings seem funny. If the left-right or up-down readings don't vary between about two and 120, chances are that you have wired the plug incorrectly. Disassemble the plug and look for loose wires or shorted wires. If you have another unmodified joystick, use it as a guide for proper wiring. □

Figure 2. Joystick Modification



The listing:

```
10 'Joystick Tester Program
20 KEY OFF:CLS
30 LOCATE 3,33:PRINT "RIGHT JOYSTICK"
40 X=5:GOSUB 90
50 LOCATE 8,33:PRINT "LEFT JOYSTICK"
60 X=11:GOSUB 90
70 F$="###"
80 GOSUB 140:GOTO 80
90 LOCATE X,10:PRINT"BUTTON 1 IS ";
100 LOCATE X,27:PRINT"BUTTON 2 IS ";
110 LOCATE X,45:PRINT"L/R=";
120 LOCATE X,60:PRINT"U/D=";
130 RETURN
140 STRIG ON
150 LOCATE 25,1:PRINT"HIT ANY KEY TO EXI
T"
160 A1=STRIG(1):B1=STRIG(3):A2=STRIG(5):
B2=STRIG(7)
170 IF A1=0 THEN A1$="OFF" ELSE A1$=" ON
"
180 IF A2=0 THEN A2$="OFF" ELSE A2$=" ON
"
190 IF B1=0 THEN B1$="OFF" ELSE B1$=" ON
"
200 IF B2=0 THEN B2$="OFF" ELSE B2$=" ON
"
210 LOCATE 5,22,0:PRINT B1$;:LOCATE 5,39
:PRINT B2$;
220 LOCATE 11,22:PRINT A1$;:LOCATE 11,39
:PRINT A2$;
230 LR1=STICK(0):UD1=STICK(1):LR2=STICK(
2):UD2=STICK(3)
240 LOCATE 5,49:PRINT USING F$;LR1;:LOCA
TE 5,64:PRINT USING F$;UD1;
250 LOCATE 11,49:PRINT USING F$;LR2;:LOC
ATE 11,64:PRINT USING F$;UD2;
260 FOR I=1 TO 100:A$=INKEY$:IF A$<>"" T
HEN END ELSE NEXT I
270 RETURN
```

PCM

A New Approach to Model 100 Binary File Exchange

By Marty Goodman and Frank Oechsly

At club meetings it often happens that Model 100 owners want to share files with each other. It is relatively easy to use a null modem cable to link the RS-232 ports of the two computers and the built-in software in the Model 100 to exchange text (.DO) files. This can be accomplished using either *Telcom* or directly from the *Text* program itself. In the former case, one merely sets up one computer to transmit and the other to receive the file. In the latter case, one first types on one computer LOAD"COM:88I1E and on the other SAVE "COM88I1E. This saves the text file in one computer via the RS-232 port into the currently opened text file area of the *Text* program of the other computer.

If you wish to exchange tokenized BASIC program (.BA) files, the approach is a little more complex. You must first convert the .BA file into a .DO file by loading it into BASIC, then saving it out again with the A option (SAVE "FNAME",A). This places a copy of the program in ASCII text form in the memory of the computer. Now you can transmit this ASCII (.DO) version of the BASIC program over to the other computer using the methods previously discussed.

The Problem with Binary Files

Binary (.CO) files present further problems. The Model 100 built-in software system is not well set up for exchanging those files. There are various utilities circulating that convert binary files to and from ASCII form (Intel Hex conversions are good for this). Once in Intel Hex or some similar ASCII form, the file can be transmitted via null modem.

But in this case both parties need extra software in their Model 100. While such utilities are written in BASIC and can be transferred themselves, if both parties do not have them in their machines at the start, the file transfer process becomes a cumbersome matter of "ASCII-fying" two BASIC programs, transmitting them, converting, then ASCII-fying the .CO program, transmitting it, and finally converting it. I've used this technique and it works, but it is a hassle.

The Technique

In such cases it was often faster to save the file to cassette tape, then reload the file into the other machine. I wondered, why not send the cassette signal directly from one machine to another? Frank Oechsly (author of the local M100/NEC 8201 newsletter) and I decided to try this out. Our first experiments with such transmission failed, because the amplitude of the signal coming out of the M100 was a bit too low for the M100 cassette signal input on the other side.

We bought a speaker/amplifier unit (Radio Shack cat. no. 277-1008). Running the output (black, ⅛-inch jack on cassette cable) of the sending M100 into the input of amplifier, and the output of the amplifier into the input of the receiving computer (gray ⅛-inch jack on cassette cable) we found transmission worked, but was unreliable because, even with the volume all the way down, the signal strength was too high.

Martin H. Goodman, M.D., a physician trained in anesthesiology, is a longtime electronics tinkerer. He is the database manager of RAINBOW's CoCo SIG on Delphi.

Frank Oechsly is the author of a California Model 100/NEC 8201 newsletter and enjoys working with portable computers.

We put a 150-ohm resistor in parallel with the input to the speaker/amplifier unit, attenuating the input signal. With this resistor added, direct transmission between Model 100s was flawless on all attempts with the volume on the amplifier box set in the low region.

Note that the speaker amplifier unit has two sockets that fit the ⅛-inch phono jacks at the ends of the M100 cassette cable. This technique allows more direct transfer of BASIC programs, as one can send the tokenized BASIC program directly from the memory of one computer to the other without first putting it in ASCII.

Discussion

Transmission via the cassette port takes place at an average of 1,500 Baud. It also includes error checking, so if an error is detected by the receiving computer, the transmission is terminated. Note that this is only error checking and (unlike XMODEM) not associated with error correcting. All the receiving computer does is cancel the transmission if it detects an error.

While you can use this technique for transferring text files as well, by saving and loading to and from cassette on the Model 100s, this becomes very inefficient. The null modem RS-232 is far superior because it works reliably at 9,600 Baud. Using the cassette port for text file transfer, you are not only tied to 1,500 Baud, but the file is transmitted with gaps between blocks of data. This approach makes the data recoverable from partly damaged tapes, but slows machine-to-machine transmission.

Try It Yourself

All you need is the Radio Shack speaker/amplifier, two Model 100s, two cassette cables, a 150-ohm resistor, a ⅛-inch mini-phono plug and ⅛-inch mono-mini socket. Make a little adaptor cable with the resistor soldered between input and ground on that cable. This will let you take the resistor in and out of the circuit as needed.

Connect the black plug from the receiving computer to one adaptor socket that has the 150-ohm resistor going to ground, and put the plug on the other side into the input jack of the speaker amplifier. Plug the gray jack from the receiving computer into the output jack of the speaker amplifier. Set the volume control to just a little beyond the "on" position.

When the computers are connected via this null cassette modem, first go to BASIC and give the receiving computer the CLOADM command if you want to receive a binary machine language program file, or give it the CLOAD command if you want to receive a tokenized BASIC program. It will wait for a signal. On the sending computer, first LOADM or LOAD the .CO or .BA file you want to transfer, then use the CSAVEM or CSAVE command to initiate transmission. Note that you must provide the needed addresses for start, end and execute if you are sending a .CO file. These addresses are displayed at the time you LOADM such a file.

When you start transmission, the receiving computer is fooled into thinking the signal is coming from a cassette recorder and CLOADs or CLOADMs the program just as if it were getting the program from the cassette recorder. After transmission is completed, you must SAVE or SAVEM the received file to put it into the directory on the receiving machine. PCM

BASIC BUSINESS™

Accounting and Business Management Software

Basic Business II MS-DOS or PC-DOS **$249.95**

Basic Business I MS-DOS, PC-DOS or CP/M **$149.95**

Advanced Color On-Screen-Help Windows

with computerized indexing into the users manual

Never again wonder what to do next, or how to do it. Basic Business is the **only** advanced accounting system that provides a complete windowed help and index system. Answers to your questions are as close as the touch of a key.

Not another cigar box system, Basic Business II offers features equivalent to those found in systems costing hundreds more. Accounting has never been easier or more complete.

Need less help? Considere the popular Basic Business I. It has most of the features found in Basic Business II without the windows and help, and is upward compatible should your needs change.

Consider the flexibility of Basic Business. You can generate over 300 reports or use the optional dBase File Formats for ultimate information gathering and reporting. Transfer information to a database or spreadsheet program for total custom reporting ability. Source code is also available for those companies requiring **customization** or having specialized accounting needs.

Minimum Hardware Requirements: 80 X 24 CRT, 2-360K floppies, 132 column printer, 64K CP/M 2.x, 128K MS-DOS or PC-DOS 2.0. PC/XT/AT or compatible.

30 day money back guarantee if all claims and features listed in this ad are not true. MasterCard and VISA welcome. Add $2.50 for shipping. CA residents add 6% sales tax.

Optional Products

Point-of-Purchase Module... $99.95

dBASE II/III File Formats ... $29.95

some disk formats slightly higher

PARTIAL LIST OF FEATURES

General Information
- All modules are menu driven / password protected
- Unlimited data entry in all modules
- Prints hundreds of detailed management reports
- Superior documentation for ease of use
- Stock or preprinted forms available
- Supports user defined notes / comments on all forms
- All modules completely integrated

General Ledger
- Unlimited user defined Chart of Accounts
- Prints all standard financial reports
- Up to 99 user-defined departments

Accounts Receivable & Billing
- Open item or balance forward
- Verifies customer credit limit at invoice entry
- Supports service or inventory billing
- Prepares customer statements with dunning notes
- Maintains full cash flow analysis
- Prepares mailing labels and customer lists

Accounts Payable
- Cash disbursement management for all vendors
- Supports hand written checks
- Automatically posts recurring invoices monthly
- User defined aging periods
- Prints vendor mailing labels and lists

Payroll & Job Costing
- Supports labor related job costing
- Maintains federal, state and local tax tables
- Handles vacation and sick leave time
- Prints W-2 forms and 941 information

Inventory
- Up to 12 digit alphanumeric part numbering
- Supports standard / average / suggested resale costing methods
- Integrated with Sales and Purchase Order systems

Purchase Order
- Maintains PO and Inventory files
- Prints confirming POs
- Prints price variance report by part number
- Maintains vendor purchase history

Order Entry & Sales Analysis
- Supports ordering and shipment of stock and non-stock items
- Handles partial shipments / cancellations / changes of any SO
- Automatic credit checking during SO entry / invoicing / billing
- Provides extensive sales analysis reporting
- Maintains MTD billing and booking totals
- Line item/total invoice discounts during SO entry and/or invoicing

Point of Purchase (optional)
- Entry of invoices posts to all pertinent modules
- Invoices stock / non-stock items
- Line item discount and tax application
- Verifies customer credit limits
- Supports cash/ check/credit card/ payment on account and returns
- Controls electronic cash drawer

Trademarks: CP/M - DRI; dBASE - Ashton-Tate; MS-DOS - Microsoft; PC-DOS - IBM

CALL TODAY National **(800) 821-8778**

California **(800) 521-7182**

Hours: Mon - Fri 8 am - 5 pm PDT

DEALER INQUIRIES WELCOME.

advent products inc.

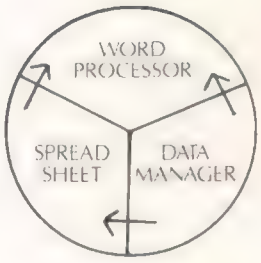
3154-F E. La Palma Ave.
Anaheim, CA 92806
(714) 630-0446

A Truly Integrated Software

NEW! MODEL 100 **3-in-1** ROM

Interactive Solutions*

All the software that you will ever need
on the Model 100 Portable:



1— Data Base
2— Spreadsheet
3— Word Processor
with Mail Merge

Now on a plug-in ROM. ***Integrated*** and easy as 1-2-3!

Interactive Solutions
Tandy Cat. #26-3844

Available at your nearest
Radio Shack
COMPUTER CENTERS

■ **The Data Base** turns your Mod 100 into a personal filing system. You can store, examine, update select, sort and list a wide variety of information - from personnel records and inventory facts to appointment schedules and client lists.

■ **The Spreadsheet** turns your Mod 100 into a powerful decision tool. It is a full featured spreadsheet with 99x99 cells and provides fast decimal arithmetic in formula computation with complete range of scientific functions.

■ **The Word Processor** uses the Text editor of Mod 100 and gives you additional features for formatting and printing. You can set margins, line/para spacing, justify, define header/footer and use special print features like underline, **boldface**, etc.

MORE MEMORY Interactive Solutions does not occupy RAM memory, so frees your precious RAM for larger data files.

EXTRA CONVENIENCE Interactive Solutions plugs in the optional ROM socket. It is always resident and ready to work. You don't need to load programs from cassette recorder any more.

Interactive Solutions *integrates* all of your data information in Mod 100. With a touch of a function key, you can easily:

- Select and move items from **Data Base** files into **Spreadsheet** columns.
- Generate a report in **Data Base** program and move to the **Word Processor**.
- Cut any part of **Spreadsheet** and move to the **Word Processor**.
- Merge the items (like names, addresses) from **Data Base** files into a **Form Letter** and generate multiple letters or labels.

All this and much more in a 32K bytes ROM from

DCM Data Products
1710, Two Tandy Center
Fort Worth, Tx. 76102
Tel: 817/390-2762

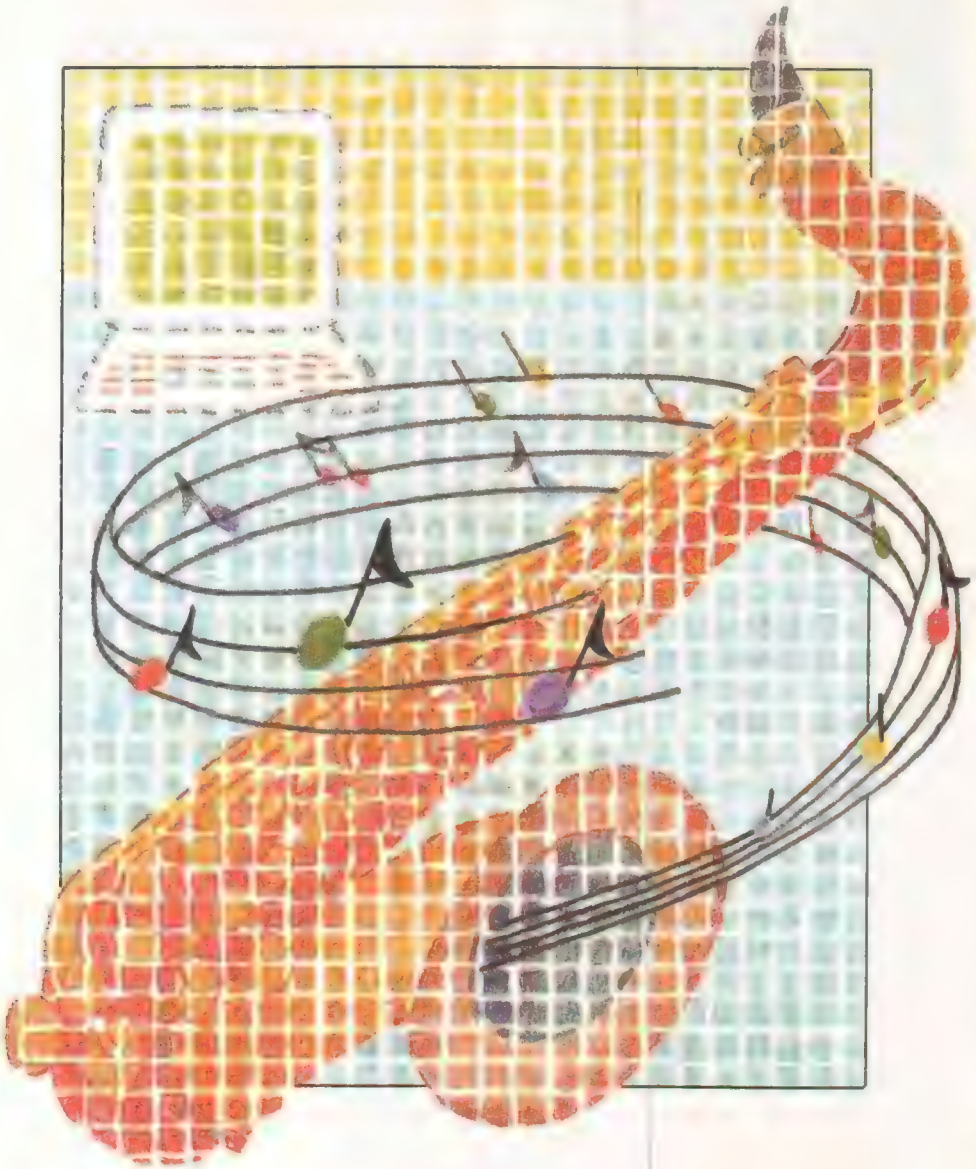
DCM data products

*Copyright DCM Data Products, Licensed to Tandy Corporation.
® Registered trademark of Tandy Corporation.
Radio Shack and Model 100 are trademarks of Tandy Corporation.
1-2-3 is a trademark of Lotus Corporation.

3-D Sounds on the Tandy 1000

By Robert W. Keefe

It is difficult to find fault with *Turbo* PASCAL from Borland International. It's easy to use, a great editor, provides painless debugging plus has extensions to the language to increase flexibility. But, when using *Turbo* on the Tandy 1000, there is one slight difficulty. Some of the nicer features of the Tandy 1000 are not easily accessed. *Turbo* thinks it is running on an IBM PC and, therefore, does not recognize the existence of some of the extensions to the IBM standard provided by the Tandy 1000. This article deals with one facet of this problem.



The Sound and NoSound statements provided by *Turbo* generate fixed amplitude signals, one frequency at a time, via an 8253 counter. There is no direct means of accessing the 76496 complex sound generator in the Tandy 1000. This complex sound generator gives the Tandy 1000 the ability to produce three independent sound channels, each with its own frequency and amplitude level. In addition, it provides another channel for noise generation.

The wizards at Borland have, however, provided us with an indirect means of utilizing these resources. *Turbo*'s Port array gives us access to all of the I/O ports and, thereby, entry into the Tandy's hardware inner sanctum. The complex sound generator may be programmed by writing data to one of these I/O ports ($00C0). The listing for *Sound_1K.inc*, which contains a group of procedures that use this method to fully utilize the sound capabilities of the Tandy 1K, is at the end of this article.

First, a few words about the 76496. This device is a programmable tone/noise generator capable of producing four sound channels simultaneously: three musical tones and one noise. The

Robert Keefe is the father of five daughters and is retired from a local power company where he was a lineman for 30 years. He is a hobbyist with a long-standing interest in all forms of electronics. Robert may be contacted at 644 Penning Avenue, Wood River, IL 62095, (618) 254-9767. (Please enclose an SASE when writing.)

Take off with a Tandy® portable and have real computing power wherever you travel.

Great traveling companions

Whether you're a businessperson, journalist, police officer or student, you can't afford idle time. Our affordable Model 100, Tandy 200, and Tandy 600 portable computers are invaluable tools that help keep you on top of your work wherever you go.

Computing power for less

The popular Model 100 (26-3802) is perfect for writing letters, reports and other correspondence anywhere, at any time. Or use the 100 as your personal appointment calendar and name/address and phone directory. This lightweight computer was the first to feature five instant-on programs, as well as a 40×8 display—all built in.

At only $499, this powerful little traveler offers you more options than you thought possible—including a built-in modem and a communications program that lets you access other computers or national information services by phone. You can even write your own programs in BASIC.

Powerful portability

The $799 Tandy 200 represents a true second generation in portable computers. Not only does it come with its own built-in software, improved features, a bigger 40×16 fliptop screen, and a larger memory, but it weighs only 4½ pounds.

Plus, the Tandy 200 (26-3860) comes with powerful Multiplan® spreadsheet analysis built right into the memory. You can do complex spreadsheet calculations at the touch of a key. Answer thousands of "what if . . . ?" questions instantly.

Our most advanced laptop

The high-performance Tandy 600 (26-3901) features a 16-bit microprocessor, an 80×16 display, a built-in 3½" disk drive that stores 360K of data, and 32K RAM (expandable to 224K).

At $1599, the Tandy 600 also includes five resident applications—all designed by Microsoft for anyone who's always on the go. With the Tandy 600's larger display and expanded memory, Multiplan can handle more information.

Using MS-Word®, you'll also have quicker access to documents and more storage capacity with the built-in disk drive. File is an electronic database for names and addresses, expenses, client billing and more. With the Telecom program and the Tandy 600's built-in modem, you can keep in touch over phone lines.

Don't leave the office without one

Tandy offers you a choice of three portable computers—one is sure to fit your needs. See them all at your local Radio Shack Computer Center!

Available at over 1200 Radio Shack Computer Centers and at participating Radio Shack stores and dealers.

Radio Shack®
COMPUTER CENTERS
A DIVISION OF TANDY CORPORATION

Send me an RSC-16 Computer Catalog.

Radio Shack, Dept. 87-A-16
300 One Tandy Center
Fort Worth, TX 76102

Name _____
Address _____
City _____
State _____
ZIP _____
Phone _____

Prices apply at Radio Shack Computer Centers and participating stores and dealers. Multiplan and MS-Word/Registered TM Microsoft Corp.


TANDY 600

amplitude of each channel may be set independently. Programming is accomplished by sending data, via I/O port $00C0, to data lines (eight bits wide) on the 76496. This data is then routed to one of eight registers on the 76496.

Addressing these internal registers is a bit tricky. If you are interested in details, consult the *Tandy 1000 Technical Reference Manual* or the *Texas Instrument Data Sheet* for this device.

Look at the first procedure in the listing, Set_Freq. This procedure accepts two arguments, Channel and Freq. The first, Channel (which must be in the range of one to three), is used to select which of the three tone channels is to be set. The second, Freq (which must be in the range of 110 to MaxInt), is used to set the frequency of the channel.

The channel frequency is determined by a 10-bit divisor (Divisor) that this procedure sends to the appropriate register on the complex sound generator. Since the data bus is only eight bits wide, it is necessary to send two bytes sequentially (Byte1 and Byte2). Byte1 contains the register address and the low-order nibble of Divisor. Byte2 contains the remaining six significant bits of Divisor.

The second procedure in the listing, Set_Volume, provides for control of the amplitude of the three tone channels and of the noise channel. Set_Volume accepts two arguments. The first, Channel (which must be in the range of one to four), is used to select which of the three tone channels is to be set. The second, Vol (which must be in the range of zero to 15), is used to set the amplitude of the selected channel. For maximum output, set Vol = 0; for minimum output, set Vol = 15. This procedure then sends the necessary data to the appropriate register on the 76496.

The third procedure, Set_Noise, is used to control the noise channel on the 76496. Set_Noise accepts two arguments. "Kind" selects the type of noise to be sounded. If Kind = 0, the output is an impulse-type noise; if Kind = 1, the output is a white noise. "Vol" is used to set the amplitude (as in Set_Volume). The pitch of the noise may be varied by changing the frequency of tone channel 3.

The last two procedures in *Sound_1K.inc* are Chord and Soundoff. Chord is used to produce a three-note musical chord and accepts three frequencies and three volume levels as arguments. Soundoff mutes all four channels. Don't forget this one — this procedure is necessary to turn off the sound output, which, if not silenced, continues until the system is reset or the power is turned off.

Also included is a short demo program to try out these routines. So, there you have it: sound in its full glory on your Tandy 1000! I will leave it to you to develop a full-fledged song-playing program. □

Listing 1:

```
{---}
Procedure Set_Freq(Channel:Byte; Freq:Integer);
 Const
  PortAddr = $00C0;
  RegAddr :Array[1..3] of Byte =($80,$A0,$C0);

 Var
  Byte1,Byte2  : Byte;
  Divisor      : Integer;

 Begin
  If Freq < 110 Then Freq := Maxint;
  Divisor := Trunc(223688.0/Freq);
  Byte1   :=  RegAddr[Channel] +(Divisor And $000F);
  Byte2   := (Divisor And $03F0) SHR 4;
  Port[PortAddr] := Byte1;
  Port[PortAddr] := Byte2;
 End; {Set_Freq}



{---}
Procedure Set_Volume(Channel,Vol:Byte);
 Const
  PortAddr = $00C0;
  RegAddr : Array[1..4] of Byte =($90,$B0,$D0,$F0);

 Var
  Bite : Byte;

 Begin
  Bite   := (Vol And $0F) Or RegAddr[Channel];
  Port[PortAddr] := Bite;
 End; {Set_Volume}

 {---}
 Procedure Set_Noise(Kind,Vol:Byte);
  Const
   PortAddr    = $00C0;
   VolRegAddr  = $F0;
   TypeRegaddr = $E0;

  Var
   Byte1 : Byte;


  Begin
   Byte1 := TypeRegAddr + ((Kind And $01) SHL 2) + $03;
   Port[PortAddr] := Byte1;
   Byte1 := VolRegAddr + (Vol And $0F);
   Port[PortAddr] := Byte1;

  End; {Set_Noise}



{---}

Procedure Chord(Freq1,Freq2,Freq3  : Integer;
                Vol1,Vol2,Vol3 : Byte);
 Begin {Chord}
  Set_Freq(1,Freq1);
  Set_Volume(1,Vol1);
  Set_Freq(2,Freq2);
  Set_Volume(2,Vol2);
  Set_Freq(3,Freq3);
  Set_Volume(3,Vol3);
 End; {Chord}



{---}
Procedure SoundOff;
 Begin
  Set_Volume(1,$0F);
  Set_Volume(2,$0F);
  Set_Volume(3,$0F);
  Set_Volume(4,$0F);
 End; {SoundOff}
```

Listing 2:

```
Program TryOut;
{R. Keefe
 Wood River,Ill.}

{$I Sound-1K.Inc }


Begin
 Chord(659,784,1047,2,2,2); delay(400); SoundOff; Delay(100);
 Chord(523,659,784,2,2,2); delay(200); SoundOff; Delay(100);
 Chord(523,659,784,2,2,2); delay(200); SoundOff; Delay(100);
 Chord(523,698,880,2,2,2); delay(400); SoundOff; Delay(100);
 Chord(523,659,784,2,2,2); delay(400); SoundOff; Delay(500);
 Chord(573,784,988,2,2,2); delay(200); SoundOff; Delay(400);
 Chord(659,784,1047,2,2,2); delay(200);
 SoundOff;
End.
```

PCM

Based on Marvin Mallon's book, this new column takes you on a grand tour of the world's most popular lap-top computer

Exploring the Tandy 100

By Marvin C. Mallon

You have just invested in a remarkable, bantam-weight microcomputer, the Tandy 100. If this is the first time out of the shipping box for your new electronic helpmate, it might be appropriate to run through the set-up procedure. A saying that dates back to the beginning of the microcomputer age states, "if all else fails, read the manual." Rather than wait for something to fail or, worse yet, for some damage to occur, you are well advised to make your way through the Tandy 100's user's manual. This owner's guide contains start-up information which can be condensed as follows:

1. Get four size AA alkaline batteries and place them in the slide-out compartment on the bottom of the machine (Figure 1).
2. Switch the "Memory Power" switch (located underneath the unit) to "On" (Figure 2).
3. Slide the main power switch to "On" (Figure 3).
4. You should get a display similar to that shown in Figure 4.

Alternatively, if you purchased the optional AC Power Adapter (part number 26-3804), plug it into the DC 6V connector as illustrated in Figure 5. This unit will save a lot of battery expense when your Tandy 100 is used at a desk and portability is not a factor. At best, the batteries will supply power for about 20 hours before replacement

Marvin Mallon is a professional microcomputer programmer and business consultant. He has written numerous technical articles dealing with a variety of computer hardware and software dating back to 1976. Mr. Mallon's book, Exploring the Radio Shack Model 100, *from which this installment is excerpted, is available from Compu-Quote located at 6914 Berquist Avenue, Canoga Park, CA 91307, (818) 348-3662.*

is necessary. Invest in plenty of extras, but use the AC Power Adapter whenever a 110 volt wall plug is handy.

On/Off and Display Controls

As you previously discovered, the On/Off switch is located on the right-hand edge of the Tandy 100. If it's in the on position, and you don't touch the keyboard for about 10 minutes, the machine will turn itself off to conserve power. It is then necessary to slide the switch back to "Off," and then to "On" the next time you wish to activate the computer. This "time-out" feature can be set for other time periods or cancelled altogether. Refer to the POWER command as explained in Chapter 9 or in the BASIC section of the owner's manual. Any program material you might have entered into the machine will be retained. This automatic "shutdown" of the Tandy 100 does not erase anything previously entered. Note that the Tandy 100 maintains the day, date and time. A small, built-in NiCad (Nickel Cadmium) backup battery keeps the clock going as well as supplying the little amount of power required by the internal memory. As long as the "Memory Power" switch is turned on, this battery will do the job until it reaches its end-of-life (approximately 2½ years). Its replacement can be easily accomplished by Radio Shack service personnel.

The "Display" control is also on the right-hand edge of the machine (Figure 3) and permits you to alter the LCD (Liquid Crystal Display) appearance to suit your taste and viewing angle. You can't hurt a thing by experimenting with various settings of this control. You will probably find that it needs to be adjusted when you move the Tandy 100 from a desk to your lap.

Setting the correct day, date and time is the next order of business and is accomplished as follows (see Figure 6):

1. Since BASIC is highlighted on the LCD and that's where you want to go, press the ENTER key.
2. Type: `DATE$="07/01/86"` ENTER (using today's date in the format *MM/DD/YY*). It is necessary to enter two digits for each of the periods; for example, July becomes `07`.
3. Type: `TIME$="14:07:30"` ENTER using the correct time in the format *HH:MM:SS* and adding twelve to the afternoon hours. Don't forget the leading zeroes as in the date example above.
4. Type: `DAY$="TUE"` ENTER using the current day of the week. Only the first three letters of the appropriate day's name should be entered.
5. Type: MENU ENTER and you'll exit BASIC and return to the main menu display.

Hereafter, your Tandy 100 will maintain the correct day, date and time except for Leap Year's February 29th. which the machine does not recognize. The procedure need not be repeated unless the internal NiCad battery requires replacement or more RAM (Random Access Memory) modules are added to the machine.

We are now ready to begin our survey of the keyboard and other important controls. Figure 7 illustrates that the keyboard occupies fully two-thirds of the face of the computer.

The QWERTY Keyboard

There are 45 keys that constitute the main typing portion of the keyboard (Figure 8). The entire alphabet as well as numbers, punctuation marks, special characters and the space bar are represented here just as they are on most typewriters. A few of the symbols and the location of others may differ from what you see on your Smith-Corona, but, by and large, this is the same keyboard you've worked with before. Be assured that you can't hurt the computer by experimenting with the keys.

A few of the symbols are notably different than those you may be familiar with. Their presence is explained principally because of their usefulness in creating BASIC programs. The less than (<) and greater than (>) symbols represent the mathematical terms. The asterisk is still an asterisk, but doubles as the sign of multiplication in BASIC programming.

QWERTY derives its name, of course, from the first six letters on the second row. Other, more efficient, keyboards have been designed (notably the Dvorak) but it seems evident that the QWERTY is too ingrained in our culture to permit variation.

As on a typical typewriter, a right and left SHIFT key can be found at the ends of the fourth row (Figure 9). Either one will put you in the capital letter mode. A "press on/press off" CAPS LOCK key serves to take you from lower- to uppercase while the ENTER key performs much the same function as a typewriter's "Carriage Return." The TAB key also acts in a manner similar to typewriter operation. Pressing it advances the cursor (the blinking black square) eight spaces to the right. After the fourth tabbing to the right, the cursor advances one line and returns to the left hand edge of the display. TAB is used only with Text, the built-in word processing program (more on that later) and in BASIC programs.

The key labeled CTRL (Figure 10), located on the left-hand edge of the third row, is never used by itself, but performs special functions in conjunction with other keys. For example, the combination of pressing the CTRL key and the letter M simultaneously will move the cursor down one line. A complete chart of control key functions can be found in Appendix A of the owner's manual.

The ESC (escape) key in the upper left-hand corner serves a limited purpose. When you press it twice in succession while in the Text mode, you will be returned to the main menu. Certain escape sequences, where this key is used in unison with others, are itemized in Appendix B of the owner's manual.

The GRPH (graph) key is located in the lower left-hand position of the keyboard. When in BASIC, it is used to display 73 different symbols when it is pressed in combination with other keys. A complete list of these graphics can be found in Appendix I of the owner's manual. The graphics symbols available include minature characters of human figures, playing card suits, arrows, checkerboard patterns and a variety of other useful, albeit unusual, pictures. They can be imbedded in BASIC programs that you create or can be conjured up by simply pressing the GRPH key and another key simultaneously. Thirty-nine symbols are available in that manner; another 34 if the SHIFT key is pressed as well.

The key labeled CODE to the right of the space bar (Figure 11) is used in the same manner as the GRPH key. It produces another 32 special characters when used in conjunction with other keys. Twenty-nine more are created when the SHIFT key is added to the combinations. The characters generated by the CODE key are principally upper and lower case foreign language vowels. This arrangement permits you

Memory Upgrades for Tandy Portables!

Model 600

96K RAM Upgrade
$249. or $465. for two

Save up to **$335.** with this upgrade designed specifically for the Model 600. The 96K upgrade card fits into existing slots in the Model 600 and adds either 96K or 192K of RAM to the computer.

Model 200

24K RAM Upgrade
$89.00
Two for $170.

Save up to **$330.** with this exact replacement while bringing your Model 200 up to 72K.

Model 100, 8201 or M10

8K RAM Upgrade
$27.00
$75. for pkg. of 3

Save up to $282. upgrading an 8K Model 100 to 32K. The modules also fit the NEC 8201 and the Olivetti M10.

★ ★ ★ BUY QUALITY & SAVE ★ ★ ★

All three upgrades feature low power version of CMOS static RAM for minimum battery drain and maximum reliability. Our products are 100% tested before they leave the factory. Every upgrade carries with it a 30 day satisfaction money back guarantee and 1 yr. replacement warranty. Step by step illustrated instructions are included, and our customer service support is as close as your phone.

TO ORDER
CALL 714-540-1174
or WRITE
M/C, VISA, CHECK or MONEY ORDER
CALIF. RESIDENTS ADD 6% Sales Tax
SHIPPING-ADD
UPS Ground $1.50
2nd Day Air $4.00
Next Day Air $12.00

Tandy is a registered trademark of Tandy Corp.

American Cryptronics, Inc.

(Formerly Cryptronics, Inc.)
1580 Corporate Drive, Suite 123
Costa Mesa, California 92626
(714) 540-1174

to compose documents in French, Swedish and other European alphabets using the Text program available in this machine. All together, some 134 symbols can be produced by combining either the GRPH or CODE key with the conventional keys previously described.

The NUM (number) key serves a special purpose. If pressed, it will stay down and convert the 10 keys above it into a miniature numeric keypad. These keys are clearly identified by the small black number on a white square in the lower right-hand corner of the keytop. While the NUM key is in this mode, numerical data entry can be speeded up by using these keys as you would on a calculator or adding machine. Pressing the NUM key again disables this feature and the ten keys revert to their conventional function on the QWERTY keyboard.

The DEL/BKSP key is the last of the full-sized keys that requires description. It is located in the upper right-hand corner of the keyboard (just below the cursor control keys) and serves double duty. You might have already discovered the backspace key acts as your electronic eraser. Each stroke causes the cursor to back up one space and "eat" the character in its path. This handy correction feature is available at all times and in all program modes. Either SHIFT key changes this into a delete key that pulls characters on the display in from the right rather than moving the cursor to the left as does the backspace key. This function is only operative, however, in the Text mode when creating or editing documents. At other times, it simply acts as does the backspace key. The combination of both the shifted and unshifted use of this single key lets you do most any editing task quickly and efficiently.

Programmable Function Keys

Located just beneath the LCD display are the eight programmable function keys (Figure 12). These keys have different meanings depending upon which application program you have selected. The Tandy 100, as witnessed by the listing of files on the main menu display, offers five choices. Each of these is discussed fully in the chapters ahead. Figure 13 enumerates the functions performed by each of these keys under each of these program selections. In addition, these keys can be redefined when the machine is operating in BASIC. This feature enables you to program as many as eight different actions that a user may select by merely pressing one of the function keys. Later, when we deal with BASIC programming, we'll explain how this is accomplished. Note Function Key 8 (F8) is reserved as the "menu" key. Regardless of the application mode you have selected, this key, when pressed, will terminate activity, preserve any files you may have created and return you to the main menu.

The Command Keys

Just to the right of the eight programmable function keys is another group of four keys that serve as special "command" keys (Figure 14). The first of these is the PASTE key. It is operational only when the machine is in the Text mode. Used in conjunction with the "Cut" function, PASTE allows text to be manipulated during document editing.

The LABEL key is used to alternately display on the bottom line of the screen the operations performed by the eight programmable function keys. As you move from application to application, the function keys (as explained previously) take on different definitions. You need only press the LABEL key to see the current definitions for these keys. Pressing it again removes this information and permits the last display line to be used for other purposes.

"If you press the PRINT key and your printer is inoperative, the computer will stall."

The PRINT key functions at all times and under all applications. When pressed, it causes the current screen display to be sent to your printer. Used in conjunction with the SHIFT key, it will print out an entire BASIC or Text file.

If you press the PRINT key and your printer is inoperative, the computer will stall. You can terminate the printing command by pressing the SHIFT-BREAK combination. While some printers will reproduce the graphic symbols that may have been created on the screen, others will either ignore them or behave peculiarly when these characters are encountered. Erratic performance of your printer can be halted by using the SHIFT-BREAK keys. Other means exist for sending information from the computer to the printer and will be discussed in future installments.

The last command key is labelled BREAK/PAUSE. If pressed while a BASIC program is being listed or run, it will halt the machine until it is pressed again. This is most useful in that it allows you to momentarily stop a display that would otherwise scroll off the top of the screen. If pressed in combination with the SHIFT key, a BASIC program will be stopped. Only a CONT or a RUN command will start the program again. This SHIFT-BREAK combination is used in other applications as well and serves to abort whatever operation may have been started. As mentioned in the previous paragraph, it is the method by which a PRINT command can be rescinded and control of the Tandy 100 returned to the operator.

Cursor Control

The last keys to be identified are the cursor control group located in the upper right-hand corner of the keyboard (Figure 15). As the arrows beneath them indicate, they are used to move the blinking square (cursor) over the face of the LCD. They are most used during word processing but also serve to move the selector bar across the screen of the main menu. Used in conjunction with both the CTRL and SHIFT keys, they perform special functions in the Text mode.

Other than the screen display, the only remaining item of interest on the face of the TRS-80 Model 100 is the "Low Battery Indicator" (Figure 16). When this LED (Light-Emitting Diode) lights up, you should replace the four AA batteries immediately. There is a reserve of about 20 minutes of operation before the computer will cease functioning. Of course, if you have an AC adapter, power is always available and no such indication will be given.

Summary

Our survey of the keyboard is complete. All 72 keys have been discussed and we should be ready to move on to some productive activity on the machine. PCM

Data Collector and Merge

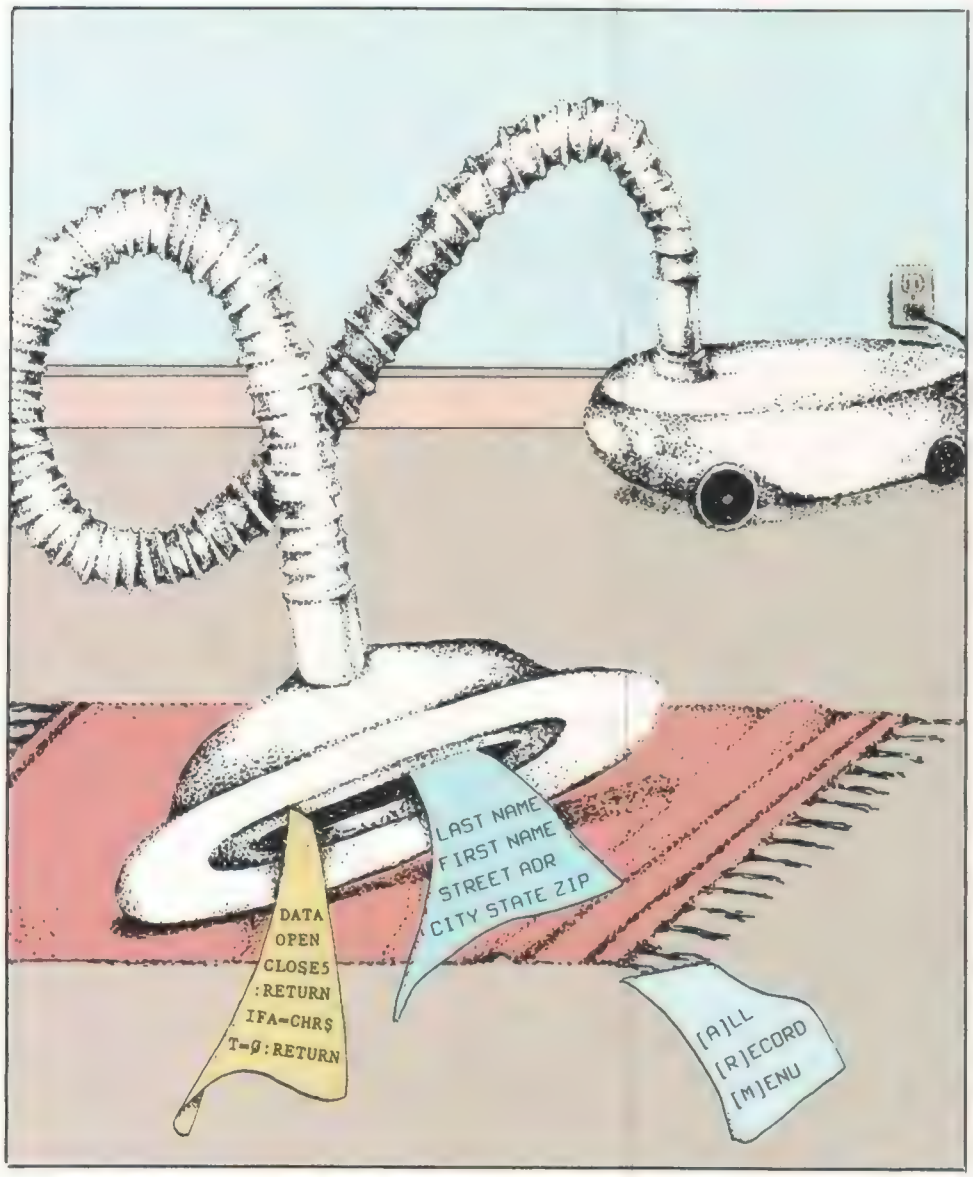
By Bennett Shulman

When the Model 100 first came out I was doing work that involved collecting public information. I had been looking for a way of collecting the information so it could be transferred to a desktop computer. Without much of an idea of exactly how I was going to do this, I bought a Model 100.

After studying the situation, I wrote a BASIC program to do this task; it was about 3,600 bytes long. After much study in program efficiency and several revisions, the latest version of this program is *Daco.100*, and is slightly over half the original length.

After I had collected the data, I was cutting and pasting parts of it into other documents with TEXT when the thought occurred to me — why not have the computer do that and print out the number of completed copies I need? To fill that need, I wrote *Datmg.100*.

These programs and others that access the database created with the use of *Daco.100* have turned out to be more than adequate for my needs. Although I do transfer the data to a desktop computer, that is done only for back-up purposes. I do all the work on the Model 100.



Program Set Up

Before using *Daco.100*, you must open two RAM files with TEXT. The first I will call the label file. This file sets the structure of the data and names each data item. The other one is the data file where the collected data will be accumulated.

Opening the Data File

You need to decide how many file headers will appear at the beginning of the file (there must be a minimum of one). Their purpose is to provide a short description of what data is in the file.

Bennett Shulman is an assistant professor at Lansing Community College and holds degrees in both law and accounting and is self-taught in the area of personal computing.

Radio Shack has the

Computer System Furniture

299.00 Reg. Separate Items 319.80

Spacious modular workstation is ideal for home or office. Features simulated oak finish. Easy to assemble.

A System Desk. Twin cable management slots for easy connections.
26-1356 . . . . . 119.95

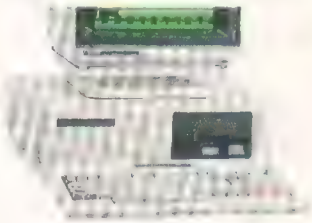
B Storage Hutch. Mounts on system desk. Features adjustable "second shelf" on either side.
26-1359 . . . . . 59.95

Half-Width Hutch. (Not shown). 26-1374 . . . 54.95

C Corner Section. Connects system desk and printer stand for a convenient "work corner."
26-1358 . . . . . 39.95

D Printer Stand. With output paper catcher and storage shelf.
26-1357 . . . . . 99.95

Monitor Pedestal



16.95

Adjustable platform rotates and tilts. Holds all Tandy monitors. Color and finish match monitors. Base: 11 × 10 1/4". (26-1369)

Printer Supports

16.95

Elevates printer so fanfold paper can be placed underneath. Use with 80-column printers. Durable crystal polystyrene. Easy to assemble. (26-1367)

Diskette Storage

Low As **12.95**

3 1/2". Holds 30.
26-1381 . . . . . 12.95

5 1/4". Holds 50.
26-1362 . . . . . 14.95

8". Holds 50.
26-4953 . . . . . 24.95

Disk Drive Head Cleaning Kits

Low As **7.95** Each

Disk Size	Cat. No.	Each
*3 1/2"	26-419	9.95
5 1/4"	26-408	7.95
8"	26-4957	8.95

*Single sided only

Fanfold Printer Paper

Size	Parts	Design	Vertical	Qty.	Cat. No	Price
14 7/8 x 11"	1	Greenbar	No	500	26-1330	11.95
14 7/8 x 11"	1	Greenbar	No	1500	72-300	34.95
14 7/8 x 11"	1	Greenbar	No	3500	26-1417	69.95
14 7/8 x 11"	1	White	No	1500	72-303	34.95
9 1/2 x 11"	1	Greenbar	Yes	1500	72-304	24 95
9 1/2 x 11"	1	Greenbar	Yes	3500	26-1403	49.95
9 1/2 x 11"	1	White	Yes	500	26-1423	8.95
9 1/2 x 11"	1	20# White	Yes	500	26-1387	10.95
9 1/2 x 11"	1	20# White	Yes	1250	26-1427	24.95
9 1/2 x 11"	1	20# White	Yes	2500	72-311	44.95
9 1/2 x 11"	2	White	Yes	750	72-305	34.95

Mailing Labels

Pressure- Sensitive Fanfold

Low As **6.95**

1-Wide. 4 1/2" carrier. Pkg./1000. 26-1328 . . . . . 6.95
1-Wide. 4 1/2" carrier. Pkg./5000. 72-402 . . . . . 24.95
2-Wide. 9 1/2" carrier. Pkg./2000. 72-401 . . . . . 15.95
3-Wide. 9 1/2" carrier. Pkg./3000. 72-400 . . . . . 16.95
2-Wide. Dry gum. 9 1/2" carrier. Pkg./2400. 26-1456 . . . 9.95

Items Shown With Optional Computer Components (Not Included).

best of everything.

Low-Cost, Compact Computer Workcenter

89.95

- **Organize Your Computer System With This Compact, Efficient Workstation**
- **Beautifully Designed and Finished to "Fit In" with Other Furniture**

Our beautifully styled, simulated oak finish workcenter features a full-width monitor shelf (41 1/2 × 14 3/4"), space for computer and printer, plus plenty of workspace. This easy-to-assemble workcenter comes with adjustable glides. (26-1350)

Perfect for Home, Office or School

Data Communications Modem

59.95

DCM 3. 300-bps modem plugs directly into modular phone outlet. DB25 and 4-pin DIN connectors. Bell 103 compatible, full duplex. FCC registered. (26-1178)

Certified Unformatted Diskettes

- **Double Density**
- **Manufactured to the Highest Standards of Excellence**
- **Buy in 10-Pack Quantities and Save**

Single Sided

Size	Tracks	Quantity	Cat. No.	Price
3 1/2"	80	3	26-415	15.95
	80	10	26-416	44.95
5 1/4"	40	1	26-305	2.19
	40	3	26-405	5.95
	40	10	26-406	15.95
8"	77	3	26-4904	10.95
	77	10	26-4906	34.95

Double Sided

Size	Tracks	Quantity	Cat. No.	Price
5 1/4"	40	3	26-411	6.95
	40	10	26-412	16.95
	80	3	26-409	8.95
	80	10	26-410	24.95
	80(high density)	3	26-421	15.95
		10	26-422	44.95
8"	77	3	26-4961	12.95
	77	10	26-4960	39.95

Radio Shack®
The Technology Store™
A DIVISION OF TANDY CORPORATION

Send me an RSC-17 Computer Catalog.

Radio Shack
Dept. 87-A-13
300 One Tandy Center
Fort Worth, Texas 76102

Name _____
Address _____
City _____
State _____ ZIP _____
Phone _____

Prices apply at Radio Shack Computer Centers and participating stores and dealers.

Each file header can be no longer than 255 characters.

Upon opening the data file, select any valid .DO filename. For our example we will call the file CUSLST.DO. The first entries in the file should be the header(s), each followed by a <CR>. Nothing else should be in the file at that point. The data collected during the operation of the program will be appended to this file or will be replacement lines for lines in this file.

For expample, if the data file has one header label it would read:

```
FILE OF CUSTOMERS
```

If the file has two header labels they would read:

```
LEADS TO FOLLOW UP
WESTERN DISTRICT
```

Creating the Label File

Next you need to decide how many data items will be collected and what those items will be called. For purposes of illustration, we will construct a customer file. The data items to be collected are last name, first name, street adr, city state ZIP, company name, tel #, date of last order, amount of last order and salutation. The first item of each record is used as the record identifier.

Once you have decided upon the number of file headers and data items, the label file can be written. For this example we will call the label file CUSLAB.DO. The first line should contain the number of headers (comma) number of data times (comma) number of the data item to be used as key reference (this data item is displayed at the top of the screen when a record is being entered). Select the data item that best helps you keep track of the record on which you are currently working.

For example, if the data file has one header, nine data items for each record and the first data item is our key reference, the first line of the label file would be: 1,9,1.

Next, you should enter the names of the individual data items each followed by a <CR>. The entire label file would look like this:

```
1,9,1
1- LAST NAME
2- FIRST NAME
3- STREET ADR
4- CITY STATE ZIP
5- COMPANY NAME
6- TEL #
7- DATE OF LAST ORDER
8- AMOUNT OF LAST ORDER
9- SALUTATION
```

The item numbers (1-, 2-, etc.) are not a necessary part of the data item label, but are useful for reference.

Running *Daco.100*

Now you are ready to run *Daco.100*. When the program is run the user is prompted with LABEL FILE AND RECORD FILENAMES? The label filename and the data filename should be entered, separated by a comma followed by a <CR> (for our example: CUSLAB, CUSLST). The program's main menu will then appear.

Menu prompts are answered with one keystroke and no <CR>. The functions are briefly described here.

1.) [F]ILE LABEL — Displays all file headers. You can check to make sure you are working on the correct data file. Pressing 'M' returns to the program's main menu.

2.) [I]NPUT NEW RECORDS — When selecting this function the screen clears and the label of the first data item appears on the screen. You can now enter the first data item. The program uses LINE INPUT so you can use commas and the data item can be 255 characters long. When you enter a <CR>, the screen clears and the next label appears. You can now enter the next data item for that record. This process continues until you have reached the end of your data item list. If you are not going to enter all the data at one time you can skip to the end of the record by entering a '[' by itself for a data item.

At the end of the data entry of each record, an option to change any entry is provided (Y/N). If 'Y' is selected the program scrolls through the data items again, this time showing what was previously entered along with the Key Ref: data item shown on the top line of the display. If 'N' is selected, the additional choice of continuing with the next new record is provided. If you are entering all data for each record at one time in the [I]nput function, all memory can be filled (about 27K in a 32K machine). However, it makes sense to leave about 6K free for use of *Datng.100*.

3.) [S]UPPLEMENT EXISTING RECORDS — Upon selecting this function the prompt to enter the desired record appears. A <CR> brings up the next record. A valid record identifier (the first data item of a record) brings up that record item. Upper- and lowercase are different characters in this program. Therefore, you must be consistent in how you enter the data. An invalid record identifier causes the program to read to the end of the data file, display the message END OF FILE and return to the main menu. You should not SHIFT-BREAK and restart the program; this causes the loss of data.

When scrolling through a record the data item label and the data currently in that item appear on the screen with the Key Ref: data item for that record on the top line. Data entered at this point replaces the existing data. Editing of existing data is not possible from the program. (You can enter the data file from TEXT to directly edit the data. Be careful not to place an *extra* <CR> in the data file as all records after it will be out of place.)

If you enter a <CR>, the data is unaffected and the program scrolls down to the next data item. A '[' alone ends modification of the record, leaving the current data item and all other data items of that record after the current item unaffected. The record ID prompt appears and operates as previously described. In effect, <CR> pages through the records and the data items.

4.) [D]ISPLAY RECORD — Displays an additional menu as follows:

```
[A]LL
[R]ECORD ID
[M]ENU
```

[A]LL displays all the record IDs on the screen; the computer beeps allowing you to press the PAUSE key to hold the screen. The [A]LL function can be used to help find a record you want to display, print or supplement. It can also operate as a checksum to make sure all the records have the correct number data items.

When you select [R]ECORD ID the prompt RECORD ID? appears. If you enter a valid ID, the computer searches for that record and beeps as the record starts to appear on the screen. The record (data labels and data) scrolls past at a moderate rate of speed. Press the PAUSE key to hold the screen. If an invalid ID is entered, the END OF FILE message appears and the [D]isplay menu rather than the main menu appears.

5.) [H]ARD COPY — When you select the [H]ARD COPY function the RECORD

PERRY COMPUTERS

CALL
1-800-248-3823

COLOR COMPUTERS	LIST PRICE	OUR PRICE
26-3127 64K Extended Color Computer 2	$199.95	$169.00
26-3131 Thinline Disk Drive 0 for Color Computer	299.95	240.00
26-3018 Extended Basic Kit	39.95	36.00
26-3030 OS-9 With Editor Assembler	69.95	59.50
26-3012 Deluxe Joystick	29.95	25.00
26-1208 CCR-81 Recorder	59.95	50.00
30001170 VIP Integrated Library	149.95	139.00
30001110 VIP Writer	69.95	59.00
30001210 Telewriter 64 Tape	49.95	42.00
30001220 Telewriter 64 Disk	59.95	49.00
30009110 Botek Interface		59.00
TANDY 3000		
25-4001 Tandy 3000 One Disk 512K	$2599.00	$1900.00
25-4010 Tandy 3000 One Disk & 20 Meg HD 512K	3599.00	2600.00
25-4030 512K to 2 Meg Expansion Board	499.00	395.00
25-4033 Math Co-Processor for 3000	399.95	339.00
25-4050 1.2 Meg Disk Drive Kit for 3000	299.95	255.00
25-4062 20 Meg Hard Disk Kit	799.00	679.00
25-4101 MS-DOS 3.1/Basic/DeskMate	99.95	85.00
TANDY 100, 200, 600		
26-3901 Tandy 600 Port Comp 32K 3½ DD	$1599.00	$1195.00
26-3904 Basic Rom for Tandy 600	129.95	110.00
26-3910 92K RAM Upgrade for Tandy 600	399.95	340.00
26-3860 Tandy 200 Port Comp 24K	799.00	650.00
26-3866 24K RAM Memory Exp Chip for 200	199.95	170.00
26-3802 Tandy 100 Port Comp 24K	499.00	425.00
26-3816 8K RAM Upgrade for Tandy 100	49.95	42.00
26-3808 Port 3½ Disk Drive for T100 T200	199.95	169.00
26-3815 Portable Disk Case	24.95	21.00
26-3805 Acoustic Coupler	39.95	34.00
26-1409 Printer Cable	14.95	12.70
26-1410 Modem Cable	19.95	17.00
MODEL 4D		
26-1070 Model 4D 64K 2 FD w/DeskMate	$1199.00	$895.00
26-1122 64K Memory Expansion	69.95	65.00
26-1123 Model 3 to Model 4 Upgrade Kit	399.00	385.00
26-1134 Model 4 HD Kit for 15 Meg	79.95	67.95
26-1530 Multiplan Model 4	199.00	169.00
26-1595 Super Scripsit Model 4	199.95	169.00
26-1608 DeskMate for Model 4	195.00	175.00
26-1635 Profile 4 Model 4	249.95	200.00
26-2216 CP/M Plus Model 4	149.00	127.00
26-2231 Double Duty Utility	69.95	58.00
MODEL 6000 COMPUTERS		
26-6021 Model 6000 512K 2 FD Comp	$4499.00	$2125.00
26-6022 Model 6000 512K 1 FD 15 Meg HD	5499.00	2975.00
26-6019 Model 6000 256K Memory Kit	199.95	185.00
26-6052 DT-100 Data Terminal	795.00	600.00
26-1245 10 Meg Disk Cartridge System	2195.00	1750.00
26-6018 6000 Interface Kit for 26-1245	119.95	102.00
25-3022 1000/1200/3000/Interface Kit for 26-1245	199.95	160.00
26-4171 35 Meg Hard Drive Primary	2995.00	2545.00
26-4172 35 Meg Hard Drive Secondary	2695.00	1895.00
26-4173 70 Meg Hard Drive Primary	4295.00	3350.00
26-4157 Installation Kit for Primary HD	349.00	295.00

RADIO SHACK PRINTERS	LIST PRICE	OUR PRICE
26-1276 DMP-105 Dot Matrix Printer	$ 199.00	$ 169.00
26-1275 TRP-100 Portable Thermal Printer	299.00	230.00
26-1280 DMP-130 100 cps Triple Mode Printer	349.95	285.00
26-1268 CGP-220 Color Ink-Jet Printer	699.00	595.00
26-2830 PC-695 8-Color Digital Plotter	799.00	680.00
26-1277 DMP-430 24 Wire Matrix Printer	899.00	635.00
26-1270 DWP-510 43 cps Daisy Wheel	1495.00	1225.00
26-1274 DMP-2100P 24 Dot Matrix Printer	1495.00	1195.00
26-1279 DMP-2200 HiSpeed Matrix Printer	1695.00	1440.00
26-1269 PT-64 Printer Controller	249.95	210.00
26-1498 SW-302 Printer Switch	119.95	100.00
26-1447 Bi-Directional Tractor for DWP 510	219.95	186.95
TANDY 1200		
25-3000 Tandy 1200 One FD & 10 Meg HD	$1999.95	$1525.00
25-3001 Tandy 1200 Two FD 256K	1499.00	1200.00
25-3044 Graphics Master	695.95	540.00
25-3061 Captain Multifunction Board	599.95	475.00
25-3020 TCS-100 Tape Cartridge System	1999.00	1555.00
25-3021 TCS-100 Interface Kit 1000/1200	149.95	120.00
25-3130 MSDOS/BASIC	89.95	76.50
TANDY 1000		
25-1000 Model 1000 128K 1 FD & DeskMate	$ 999.95	$ 705.00
25-1001 Model 1000 256K 1 FD 10 Meg HD	1999.00	1475.00
25-1003 300-Baud Modem Board 1000/1200	149.95	125.00
25-1005 Disk Drive Expansion 1000	199.95	170.00
25-1014 RS-232C Plus Interface 1000/1200	99.95	85.00
25-1013 1200-Baud Modem Board 1000/1200	299.95	230.00
25-1025 10 Meg Hard Disk Drive	699.95	559.00
25-1007 Hard Disk Control Board	299.95	255.00
30051030 PJB Multifunction Board 512K		310.00
MONITORS AND DISPLAY ADAPTERS		
25-1020 VM-4 Monochrome Monitor	$149.95	$125.00
26-5111 VM-1 Monochrome Monitor	199.95	165.00
25-3010 VM-3 Green Monitor	219.95	185.00
25-1021 CM-4 Color Monitor	299.95	225.00
25-1022 CM-10 Color Monitor	459.95	380.00
26-5112 CM-1 Color Monitor	599.95	475.00
25-3043 Graphics Display Adapter	299.00	185.00
25-3046 Deluxe Text Display Adapter	249.95	199.00
26-3047 Deluxe Graphics Display Adapter	499.95	295.00
EPSON PRINTERS		
20001015 Homewriter 10	$295.00	$215.00
20001025 LX-80 Dot Matrix Printer	329.00	250.00
20001035 FX-85 Dot Matrix Printer	549.00	385.00
20001050 HI-80 4 Pin Plotter	599.00	390.00
20002010 RX-100 Dot Matrix Printer	499.00	400.00
20002030 FX-286 Dot Matrix Printer	799.00	550.00
20003020 LQ-800 24 Pin Head Dot Matrix	799.00	565.00
20003030 LQ-1000 24 Pin Head Dot Matrix	995.00	695.00
10081010 DX-10 Daisy Wheel Printer	299.00	255.00
10081020 DX-20 Daisy Wheel Printer	459.00	390.00
10081030 DX-35 Daisy Wheel Printer	899.00	765.00

For Technical Questions and Information on our complete line of computer accessories and current prices:

CALL 1-517-625-4161
FOR ORDERS ONLY CALL 1-800-248-3823

Mon., Wed. & Fri. 9-9, Tues. & Thurs. 9-6, Sat. 9-3

124 S. MAIN ST, PERRY, MICH. 48872

All prices and offers may be changed or withdrawn without notice. Advertised prices are cash prices. C.O.D. accepted ($10.00 charge per carton on C.O.D. Call for further C.O.D. information.) M.C., Visa, add 2%. AX, add 3%. All non-defective items returned will be subject to 10% restocking fee. Defective items require return merchandise authorization. Call for R.M.A. Number before returning. Delivery is subject to product availability.

ID? prompt appears. If you enter a valid ID that record is printed along with the labels identifying each data item. At the end of the record a form feed, CHR$(12), is sent to the printer. If your printer uses a different control character, you can change variable M in Line 1 of the program. When the printout is complete or the end of file is reached the main menu reappears. If you attempt to print a record without the printer hooked up, the computer appears to lock up. To recover: Press SHIFT BREAK to break out of the program and then F4 to restart.

6.) [E]XIT FROM PROGRAM — Returns to the Model 100 main menu. Maxfiles will be set at two. You should exit from the program this way rather than breaking out of it and pressing F8 because data might be lost.

Correcting Errors

There are three possible sources of errors that might occur. They are: 1) Entering data with TEXT in a manner that changes the number of carriage returns in a data file (EF Error); 2) Running out of string space claimed in the CLEAR (OS Error); and 3) Running out of memory when using the [S]upplement function (OM Error).

Correcting the first two problems is easy. If the incorrect number of carriage returns is in the file, correct it with TEXT and use the display [A]LL to make sure everything is back in order. For the second, edit Line 1 of the program to increase string space claimed in the CLEAR command.

Correcting the third error is more complex. Model 100 BASIC has only sequential file access. Most disk-based computers have random-direct file access. For the Model 100 this means that in order to insert data into a RAM file as *Daco.100* does, the file needs to be copied from the top down to the point that the replacement is to be made. This replacement is actually a line substitution.

To successfully insert a replacement line, there must be two copies of the data file memory, the old one and the new updated one. *Daco.100* uses approximately 2K for the program and the file label. In an otherwise empty 32K machine this leaves 27K for data storage. If data is to be inserted in an existing 15K data file, the file, while it is being copied, will consume 30K (15K*2) and cause an out-of-memory error.

The solution is to break your data files into smaller ones (but you can keep more than one in memory at a time. We can break our 15K data file down into three 5K data files. When we insert data into one of the 5K files, memory arithmetic would be 5K for the old copy of the data file being updated, 5K for the new copy, 10K for the other two data files that are not currently being accessed and 2K for the program. This all adds up to only 22K.

The point, therefore, is to have several smaller data files rather than one large one. It is best to start out with this plan. If you end up with a large data file that you still need to supplement, *carefully* cut and paste it into at least three smaller files and add the correct number of file headers so the program correctly reads the new files.

If the out-of-memory error occurs during the update of an existing data file with the [S]upplement function, your newly entered data merged with the existing data is in a file called Z.DO. However, this file is not complete since the copying process was interrupted. Look at the end of this file and delete any incomplete records with TEXT. You must now patch the balance of your data file to Z.DO.

"The program allows forced page breaks with the inclusion of the form feed character recognized by your printer."

Return to the Model 100 menu and enter your data file with TEXT. Cut the data file with the F7 F6 from the top down to the point where entries are Z.DO stop. Cut and paste the balance of the data file to the end of the Z.DO file. Now your data file should be complete again. Kill your now empty data file with KILL"*filename*" in BASIC and name Z.DO as your data file (also in BASIC). Do the checksum to make sure everything is in order and you are set to go again.

***Datmg.100* Capabilities**

This program takes the data accumulated from using *Daco.100* and merges any data items from any records selected into a form we create in RAM with TEXT. Any number of copies of the completed form can be printed. There is an option of saving one copy of each completed form on cassette tape. The data file can be entered from RAM or cassette tape. Obviously, there cannot be both tape input and output in the same run.

***Datmg.100* Set Up**

Prior to running the program you must arrange the input and output devices (printer, and input or output, data file in RAM or tape and label file in RAM.

The data file used must be one created by *Daco.100*. It can be in RAM or on tape, but having the data file in RAM has two advantages: 1) faster execution and 2) use of the program's second pass search if the records selected are listed in a different order than they appear in the RAM data file. On the other hand, having the data file on tape uses no memory.

The label file must be in RAM and must be the same one that was used with *Daco.100* to create the data file.

The user must also open a RAM file with the name JOBSPC.DO. This file must contain the record ID of each record that will be accessed in the data merge run. The ID is the first data item of each record. Each record ID should be followed by a comma (,) with the number of copies desired, then followed by a <CR>. For example:

```
smith,2
Jones,1
BROWN,3
```

The program forces uppercase for internal use in the ID so records are not missed for this reason.

If the data file is in RAM, it will not matter if the records requested are out of order as the program makes two passes through the data before giving up. If the input data file is on tape, be sure the records in JOBSPC.DO are in the same order that they appear in the data file. The program reads to the end of the file looking for the record. The only way to be sure if this has occurred is to let the tape run to the end.

You must also create the blank form in a RAM file using TEXT; the form can have any format. The program takes care of left and right margins. You can specify these during the start up of the program. You should include the desired number of <CR>s at the top of the form to specify the top margin.

The program allows forced page breaks with the inclusion of the form feed character recognized by your prin-

ter. This can be specified during startup. The program assumes you are using paper with 66 lines. A page break is automatically generated after printing 55 lines. The program takes care of page breaks at the end of each document, so do *not* insert the form feed CHR$ at the end of the document.

In order to indicate where data is to be inserted use the down-arrow graphics symbol, '↓', for example, "↓4," selects the city-state-ZIP data item to be inserted at that point. None of the three characters take up space in the completed form, so format your form accordingly.

The print editor in the program supports non-printing control characters for such printer functions as compressed font, double-strike printing, underlining, etc. You should include such special characters (CHR$) in the form with CHR$(12). Consult your printer manual for the available functions and codes.

Running *Datmg.100*

The program has been designed to allow you to start it and let it run unattended. It first prompts you to enter the label and data filenames. They should be entered in the same manner as in *Daco.100*. Next, you are prompted to enter the form filename.

The printing parameters are the next input. Each has a default value indicated. All on the same screen, you should enter: left margin, right margin, single- or double-spaced, ASC# of the form feed character of the printer used, and tape output or not.

The screen then clears with only RECORDS NOT FOUND: remaining on the screen. The IDs of those records not located are printed on the screen as a run report. At this point, there is a pause of several seconds to a minute and a half long as the program searches for the first record specified in JOBSPC.DO and merges the selected data items into the form.

When the merging is done, the program edits the completed form to the printer, RAM and tape if requested (if a tape copy is being made each form merged will be in a separate file with the filename CAS:C). The program pauses after each line to edit the next line. The printing of additional copies after the first will be faster. The merged and edited RAM copy of the form is dumped to the printer. There is no pause for editing.

When all forms have been printed, the RECORDS NOT FOUND report is on the screen. The two copies of the form created by the program will have been killed and Maxfiles set to zero.

Datmg.100 is a flexible program because it allows you to define the blank form in any format. What may not be apparent is that your data item length as collected by *Daco.100* can be up to 255 characters. *Datmg.100* merges that entire data item where indicated and edits the line so as not to break words in the middle. □

Listing 1:

```
Ø ' DATMG.1ØØ BY BENNETT D. SHULMAN COPY
RIGHT 1984 ALL RIGHTS RESERVED
1 GOTO37
2 OPEN"E.DO"FORINPUTAS4
3 IFEOF(4)THENCLOSE4:RETURNELSELPRINTINP
UT$(1,4);:GOTO3
4 OPENEFORINPUTAS3:OPEN"M.DO"FOROUTPUTAS
5
5 IFEOF(3)THENCLOSE3:CLOSE5:RETURN
6 A=INPUT$(1,3)
7 IFA=CHR$(153)THENINPUT#3,W:PRINT#5,B(W
);ELSEPRINT#5,A;
8 GOTO5
9 OPEN"M.DO"FORINPUTAS5:OPEN"E.DO"FOROUT
PUTAS4:IFTP=1THENOPEN"CAS:C.DO"FOROUTPUT
AS6
1Ø FORQ=1TOV+1-LEN(I)
11 IFEOF(5)THENG=I+CHR$(P):GOSUB25:GOSUB
21:CLOSE4:CLOSE5:CLOSE6:I="":T=Ø:RETURN
12 A=INPUT$(1,5)
13 IFA=CHR$(1Ø)THEN11
14 IFA=CHR$(13)THENG=I:GOSUB25:GOSUB21:I
="":GOTO1Ø
15 IFA=CHR$(P)THENG=I+A:GOSUB25:GOSUB21:
I="":A="":T=Ø:GOTO1Ø
16 I=I+A
17 NEXT
18 FORQ=LEN(I)TOØSTEP-1
19 IFMID$(I,Q,1)=" "THENG=LEFT$(I,Q-1):I
=RIGHT$(I,LEN(I)-Q):GOSUB25:GOSUB21:GOTO
1Ø:GOSUB25:GOSUB21:GOTO1Ø
2Ø NEXT
21 LPRINTSPACE$(S);G:PRINT#4,SPACE$(S);G
:T=T+1:IFT>55THENLPRINTCHR$(P);:PRINT#4,
CHR$(P);:T=Ø
22 IFTP=1THENPRINT#6,SPACE$(S);G:IFT>=56
THENPRINT#6,CHR$(P);
23 IFZ=1ANDT=<56THENLPRINT:PRINT#4,:T=T+
1:IFTP=1THENPRINT#6,
24 G="":RETURN
25 IFLEFT$(G,3)="   "THENRETURN
26 IFLEFT$(G,1)=" "THENG=RIGHT$(G,LEN(G)
-1):GOTO26ELSERETURN
27 IFEOF(2)THENKILL"M.DO":KILL"E.DO":MAX
FILES=Ø:END
28 INPUT#2,F,M:K=F:GOSUB31:F=K
29 IFEOF(1)ANDX=1THENPRINTF;" ";:CLOSE1:
GOSUB34:X=Ø:RETURNELSEIFEOF(1)THENCLOSE1
:GOSUB34:X=1:GOTO29
3Ø FORQ=1TON:LINEINPUT#1,B(Q):NEXT:K=B(1
):GOSUB31:AA=K:IFAA=FTHENO=1:X=Ø:RETURNE
LSE29
31 FORZ1=1TOLEN(K):Z2=ASC(MID$(K,Z1,1))
32 IFZ2>96ANDZ2<123THENMID$(K,Z1,1)=CHR$
(Z2-32)
33 NEXT:RETURN
34 OPENDFORINPUTAS1:FORQ=1TOU:LINEINPUT#
1,H(Q):NEXT:RETURN
35 GOSUB27:IFO=1THENO=Ø:GOSUB4:GOSUB9:IF
M=1THEN35ELSEFORQ=2TOM:GOSUB2:NEXT
36 GOTO35
37 CLS:CLEAR17ØØ:MAXFILES=6:DEFSTRA-K:DE
FINTL-Z:PRINT:FILES:INPUT"LABLE FILE AND
 DATA FILE NAMES          ";LF$,D:CLS:PRI
NT:FILES:INPUT"FORM FILE NAME";E:OPENLF$
FORINPUTAS1
```

```
38 INPUT#1,U,N,R:DIMB(N),H(U):CLOSE:S=1Ø
:Y=7Ø:CLS:PRINT:INPUT"LEFT MARGIN <CR>=1
Ø";S:INPUT"RIGHT MARGIN <CR>=7Ø";Y:V=Y-S
:INPUT"[S]INGLE OR [D]OUBLE SPACE <CR>=[
S]";J:IFJ="D"ORJ="d"THENZ=1
39 OPEN"JOBSPC.DO"FORINPUTAS2:GOSUB34
4Ø P=12:INPUT"FORM FEED CHR$ ASC# <CR>=1
2";P
41 INPUT"DO YOU WANT CASSETT OUTPUT Y/N
<CR>=N";C:IFC="Y"ORC="y"THENTP=1
42 CLS:PRINT"RECORDS NOT FOUND: ";
43 GOTO35
```

Listing 2:

```
Ø '  DATA COLLECTOR  -- DACO.1ØØ -- BY B
ENNETT SHULMAN.  COPYRIGHT 1984
1 CLEAR12ØØ:MAXFILES=2:DEFINTL-V:DEFSTRA
-K:M=12:D=".DO":CLS:PRINT:FILES:INPUT"LA
BEL FILE AND RECORD FILE NAMES         ";
G,J:IFRIGHT$(J,3)<>DTHENJ=J+D:OPENGFORIN
PUTAS1:INPUT#1,T,L,S:DIMA(L),C(L),F(T):F
ORN=1TOL:LINEINPUT#1,C(N):NEXT:CLOSE:CLS
:GOSUB68
2 H=INKEY$:IFH=""THEN2
3 ONINSTR("FISDHEfisdhe",H)+1GOSUB4,6,1Ø
,21,28,56,5,6,1Ø,21,28,56,5
4 H="":GOTO2
5 MENU
6 GOSUB52:CLS:PRINT:FORQ=1TOT:PRINTF(Q):
NEXT
7 PRINT@287,"[M]ENU";
8 I=INKEY$:IFI=""THEN8
9 IFI="M"ORI="m"THENCLOSE:GOSUB68:RETURN
ELSE8
1Ø OPENJFORAPPENDAS1
11 CLS:GOSUB45
12 PRINT:PRINT"DO YOU WANT TO CHANGE ANY
 ENTRY  Y/N
13 H=INPUT$(1)
14 IFH="Y"ORH="y"THENU=1:GOSUB45:GOTO12
15 V=1:GOSUB43
16 PRINT@247,"[M]ENU";:PRINT@287,"<SPACE
 BAR> TO CONTINUE";
17 H=INKEY$:IFH=""THEN17
18 IFH="M"ORH="m"THENCLOSE:GOSUB68:RETUR
N
19 IFH<>CHR$(32)THEN17
2Ø GOTO11
21 GOSUB52:OPEN"Z.DO"FOROUTPUTAS2:FORN=1
TOT:PRINT#2,F(N):NEXT:V=2
22 U=1:IFEOF(1)THENGOSUB74:GOSUB68:RETUR
N
23 K="":CLS:PRINT:INPUT"RECORD ID";K
24 IFK=""THENGOSUB54:GOSUB45:GOSUB43:GOT
O22
25 IFEOF(1)THENGOSUB74:GOSUB68:RETURN
26 GOSUB54:IFK<>A(1)THENGOSUB43:GOTO25
27 GOSUB45:GOSUB43:GOTO22
28 CLS:PRINT:PRINT"DISPLAY MODULE":PRINT
TAB(7);"[A]LL RECORD ID'S":PRINTTAB(7);"
[R]ECORD ID ?":PRINTTAB(7);"[M]ENU";
3Ø H=INKEY$:IFH=""THEN3Ø
31 IFH="A"ORH="a"THEN35
32 IFH="R"ORH="r"THEN39
33 IFH="M"ORH="m"THENCLOSE:GOSUB68:RETUR
N
34 GOTO3Ø
35 CLS:PRINT:GOSUB52
36 IFEOF(1)THENBEEP:FORN=ØTO75Ø:NEXT:GOS
UB75:CLOSE:GOTO28
37 GOSUB54:PRINTA(1);" ";:IFPOS(Ø)>3ØTHE
NPRINT
38 GOTO36
39 GOSUB52:GOSUB69
4Ø IFA(1)=""THEN28ELSEIFK=A(1)THEN41ELSE
28
41 CLS:BEEP:PRINT:FORN=1TOL:PRINTC(N):PR
INTTAB(3);A(N):FORO=ØTO6ØØ:NEXT:NEXT:GOS
UB75
42 CLOSE:GOTO28
43 FORN=1TOL:PRINT#V,A(N):A(N)="":NEXT:R
ETURN
45 FORN=1TOL:CLS:PRINT@Ø,"KEY REF: ";A(S
):PRINT@85,C(N):PRINTA(N)
46 PRINT:LINEINPUTE
47 IFE="["ANDU=1THENCLS:GOTO51
48 IFE="["THENFORR=NTOL:A(R)="":NEXT:CLS
:GOTO51
49 IFE<>""THENA(N)=E
5Ø CLS:NEXT
51 U=Ø:RETURN
52 OPENJFORINPUTAS1
53 FORN=1TOT:LINEINPUT#1,F(N):NEXT:RETUR
N
54 FORN=1TOL:LINEINPUT#1,A(N):NEXT:RETUR
N
55 PRINT:PRINT"END OF FILE":BEEP:BEEP:FO
RN=1TO1E3:NEXT:RETURN
56 GOSUB52:GOSUB69
58 IFK=A(1)THEN59ELSEGOSUB68:RETURN
59 FORN=1TOL:LPRINTSPACE$(1Ø);C(N):LPRIN
TSPACE$(15);:FORP=1TOLEN(A(N)):B=MID$(A(
N),P,1):LPRINTB;
64 IFB=CHR$(32)ANDLPOS(Ø)>65THENLPRINT:L
PRINTSPACE$(15);
65 NEXT
66 LPRINT:LPRINTSPACE$(15):NEXT:LPRINTCH
R$(M):CLOSE:GOSUB75:GOSUB68:RETURN
68 CLS:PRINT:PRINT"   MAIN MENU":PRINTTA
B(7);"[F]ILE LABEL":PRINTTAB(7);"[I]PUT
NEW RECORDS":PRINTTAB(7);"[S]UPPLEMENT E
XISTING RECORDS":PRINTTAB(7);"[D]ISPLAY
RECORD":PRINTTAB(7);"[H]ARD COPY":PRINTT
AB(7);"[E]XIT FROM PROGRAM";:RETURN
69 CLS:PRINT:INPUT"RECORD ID";K
7Ø IFEOF(1)THENCLOSE:GOSUB55:GOSUB75:RET
URN
71 GOSUB54
72 IFK=A(1)THENRETURN
73 GOTO7Ø
74 KILLJ:NAME"Z.DO"ASJ:GOSUB55:RETURN
75 FORN=1TOL:A(N)="":NEXT:RETURN
```

PCM

The Complete Guide to PCM's Third Year

Our Annual Index

Compiled by Leslie A. Foster

INDEX TO PCM
July 1985 - June 1986

This is the second index to PCM and covers the period July 1985 to June 1986. The only change from last year is that items that are contained in PCM ON DISK are indicated with a • and are also listed at the end of the index, arranged by program name. Items that have bar coded listings are indicated with a ★, and are also listed separately.

The number of items in each subject heading is listed below—the number in brackets indicates the total number of items published since July 1983.

GENERAL - 19 (36)
MSDOS - 99 (159)
MSDOS-REVIEW 71 (97)
PORTABLE - 28 (172)
PORTABLE-EDUCATION - 0 (5)
PORTABLE-GAME - 0 (13)
PORTABLE-REVIEW - 19 (75)

TOTAL NUMBER OF ARTICLES - 236 (557)

PERSONAL AUTHORS - 144 (375)
BAR CODE LISTINGS - 20 (47)
PCM ON DISK - 47 (47)

Leslie Foster is a librarian with Dalhousie Law Library in Halifax, Nova Scotia. In August 1985, Radio Shack lent him 12 Model 100s to aid in reconstructing the library after a fire.

GENERAL

MSDOS

TEXT POWER™ 100/102

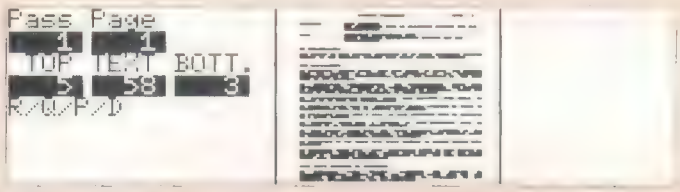
Portable 100 Magazine: ... "THE BEST FORMATTER THERE IS FOR THE 100" ... is even better than ever!

- After more than 2 years, still the most compact and still the best text formatter! Packed with powerful features not found in word processors on some desktop computers. TEXT POWER uses less than 2.7K of precious RAM, and is written 100% in machine language for speed and efficiency.
- Now available in both parellel and serial versions, with optional support for International/French characters.
- Optional compatibility for Disk Video Interface, Tandy portable drive (FLOPPY.CO) or Chipmunk drive.
- TEXT POWER supports the following features: Footers and Headers that can be turned on/off or changed from within text, Page numbering, Tab characters, MERGE TEXT™, Right Justification, Centering, Left & Right margin control with "in-file" commands, and much, much more...
- TEXT POWER also supports the following printer independant, easy-to-use embedded control commands:

^ B Boldface		^ F Near Letter Quality	10 cpi	^ U Undeline	
^ C Condensed	17cpi	^ G Near Letter Quality	12 cpi	^ W Wide/Extended	
^ D Double Strike		^ H Super Script		^ N Microfont	
^ E Elite	12 cpi	^ L Sub Script		^ R Red or Italics	

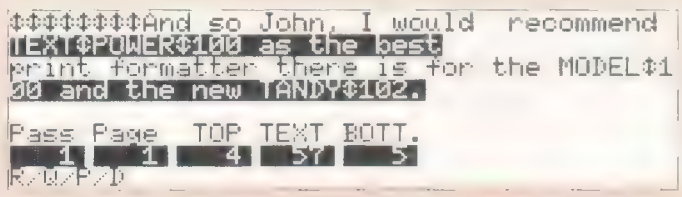
- **SET MENU**

At the stroke of a single function key, each of the displayed vital parameters can easily be modified. There is no need to reset parameters each time a document is printed, since TEXT POWER "remembers" the previous values stored. And TEXT POWER conveniently sets default parameters for standard usage.



- **PAGE PLOT™**

The most powerful and time-saving feature that exists in text formatters for portable computers today! You can graphically see your formatted text before printing! And Hugo Ferreyra set the standard by inventing and implementing PAGE PLOT™ first in TEXT POWER 100, before Write-Rom, Lapword, T-Word or any others!!!



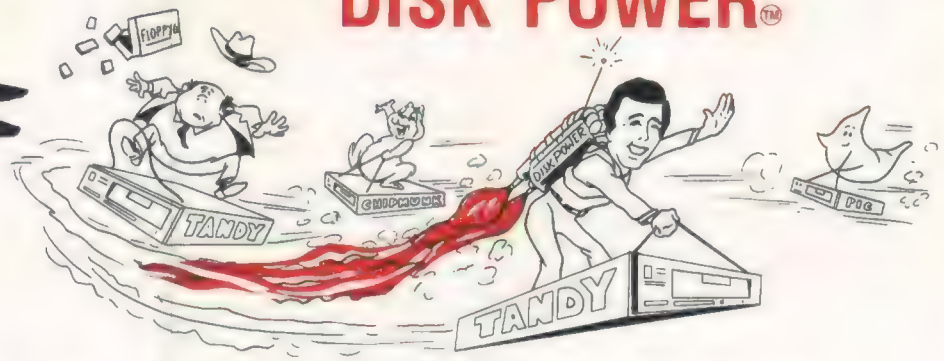
- **DISPLAY FEATURE**

Before printing, view all the headers, footers, tabs, centering, merges, page numbers, and the actual formatted text without any of the embedded commands. Scrolling can be halted or resumed at any point, and important format information is shown at the end of every page.

Now even better. Improvements in TEXT POWER 100 Version 1.42 over Version 1.38:

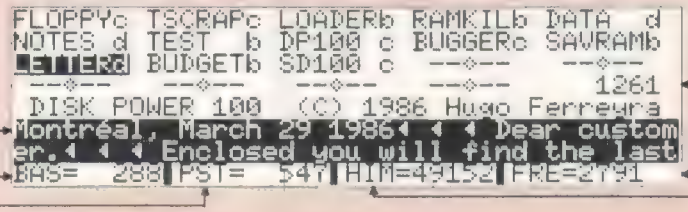
- TABs are fully supported, shown during plot and replaced with the correct number of non-breakable spaces.
- TEXT files can be accessed or created from within TEXT POWER for increased versatility.
- FORM FEED your printer directly from within TEXT POWER's Main Menu by pressing the (LABEL) key.
- PROGRAM FLOW is even more consistent for all functions, with vital parameters displayed after each page.
- PASS COUNTER implemented to help recognize which copy is being processed when merge is in use.
- DOUBLE SPACE now selected as an "in file" command to turn the feature ON or OFF anywhere during printing.
- FOOTER & HEADER handling is simplified & easier to use.
- "Plot" is improved and now shows page boundaries when there is room on the screen.
- (.DO) files only are listed when entering any function, with a default filename proposed to save typing.
- START & STOP file processing anywhere within the text, using this embedded command.
- LINE FEED support is now set internally within TEXT POWER. No need to rely on external utilities like FIXLF.
- (ESC) in almost every situation can be used to abort any TEXT POWER function.
- ERROR HANDLING is improved. For example, PRINTER NOT READY error does not "HANG-UP" your Model-100 but instead beeps to warn you of the situation.

DISK POWER™



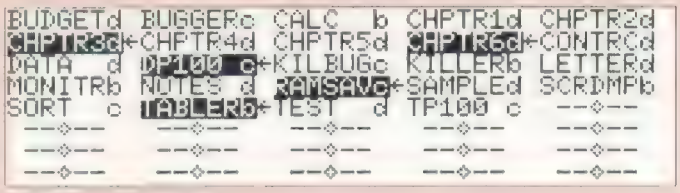
DISK POWER 100/102/200 is a stand-alone Disk Operating System for the Tandy Portable Disk Drive that completely replaces FLOPPY.CO, and comes with an extension to BASIC which can be loaded only when needed. It has operational & size advantages over any other operating system on the market.

- **RAM DIRECTORY:**
  - 80 character "WINDOW" feature for .DO or .BA files. Also displays TOP, END, EXE addresses for .CO files.
  - Number of bytes occupied by an unsaved BASIC program.
  - Number of characters in the PASTE buffer.



- File length of highlighted file.
- 7 lines of actual text packed into the two line window. ◂◂◂
- Total FREE bytes available.
- Value of HIMEM

- **DISK DIRECTORY:**
  - —In both directories, filename extensions replaced by a single lowercase character. No waste of screen space on redundant information. This allows all 40 disk files to be displayed at once! No scrolling necessary.
  - —Exclusive multiple file selection feature via single keystroke, to perform any task on upto 20 files at once!
  - —"ALL" function key automatically selects as many as 20 files, beginning at the current cursor position, to allow Load "ALL", Save "ALL", Kill "ALL", etc.



- **EXTENSION TO BASIC:**
  - OPEN disk data (.DO) files for input, output or append. Commands such as EOF, Print, Input are implemented. Full support for Load, Save, LFiles, Merge, etc.
  - All the extended functions of DISK POWER BASIC use similar syntax to existing commands, so that your favorite BASIC programs could be converted to take advantage of the Portable Disk Drive in seconds!
  - DISK POWER provides built-in protection against system crashes when users try to perform two different disk functions at the same time!

- **EXTENSION TO TEXT:**
  - —Load & Save disk .DO files from within TEXT and TEXT POWER, using the familiar F2 & F3 function keys.
- **EXTENSION TO TELCOM:**
  - —Direct access to BASIC & TEXT from inside TELCOM. Kill, Load, Save or Edit files without losing telephone communications. You can even see your files while uploading. With DISK POWER, it is all possible!
- **24 HR. BBS SUPPORT HOTLINE:**
  - —Using STAT: M8N1E, ID: Disk, PW: Power (514) 457-2917

DISK POWER is the only Disk Operating System which provides all of these INNOVATIVE features. Written 100% in machine-language to make it ULTRA-fast.

TEXT POWER 100/102 $49.95 US **DISK POWER 100/102/200 $49.95 US**

or SAVE by buying the ***ULTRA* POWER** program pack: DISK POWER/TEXT POWER 100 FOR ONLY **$89.95** US

To order *TEXT POWER* , specify configuration and printer(s) used. The basic price includes support for one printer. Please add $10.00 for each additional printer you wish supported. All U.S. orders shipped from within the USA. Absolutely no custom forms, sales tax or duty required. Add $3.00 S&H to each order. Send your original cassette to the address below, and receive an updated version of **Text Power** (V. 1.42) for only $24.95 U.S. plus S & H.

No surcharge for VISA or MC To order, call **TOLL FREE:**

USA (Except N.Y.) **1-800-448-4511 Ext. 313** NEW YORK STATE ONLY **1-800-962-1480 Ext. 313**

ONTARIO & QUEBEC **1-800-255-1133 Ext. 313**

or send CHECK or MONEY ORDER to:

ULTRASOFT INNOVATIONS

P.O. Box 219, Ste. Anne de Bellevue, Québec
CANADA H9X 3R9 (514) 457-9293 (9-5 EST)

Ultrasoft and Ultrasoft logos are registered trademarks of Ultrasoft Innovations Inc. Other Trade Marks: Tandy - Tandy Corp., Write-Rom - P.C.S.G. INC., T-Word - Travelling Software Inc., Lapword - Jim Irwin, Chipmunk - Holmes Eng.

MSDOS - REVIEW

PORTABLE

Ramella, Richard. "Sardine." *(1986, March)* 22★ — Pack programs to conserve space.
Scerbo, Fred. "Portable multiple choice test generator." *(1985, August)* 66★
Teague, John F. "Screen to memory to screen." *(1985, September)* 59★ — Store and retrieve graphics.

PORTABLE - REVIEW

"Bar code drivers." *(1985, September)* 85
"BUSS." *(1986, May)* 113
"ByteFyter." *(1985, August)* 83
"C-Num and C-Sort." *(1986, April)* 107
"DATA-SORT." *(1985, August)* 79
"DO4MAT." *(1985, December)* 79
"Exploring the Radio Shack Model 100 (book)." *(1985, December)* 84
"FAST." *(1986, May)* 117
"Interactive Solutions." *(1986, March)* 113
"Let's play Monopoly." *(1985, July)* 79
"Men-U-Tility." *(1985, August)* 83
"NavPln and NavAid." *(1985, July)* 80
"Portable light." *(1986, May)* 117
"Project scheduler." *(1985, October)* 84
"Supera." *(1985, November)* 83
"Tandy Portable Disk Drive." *(1986, February)* 124
"Textpro." *(1985, November)* 83
"TMPC." *(1986, April)* 114
"X-TEL." *(1986, June)* 116

Author Index

Megabytes for Mini-Bucks

Tandy 1000 Hard Drives

by Hard Drive Specialist

Shop and compare. Hard Drive Specialist has been building hard drive systems for years and have sold thousands of subsystems to satisfied Radio Shack/Tandy customers. We use the latest state-of-the-art disk drives and controllers. Our drives all use buffered seek logic and plated media to result in almost one-fourth the average access times found on our competitors drives. Plated media results in longer platter life and high resistance to head crashes not found on coated media drives. We utilize high quality construction throughout including heavy duty power supplies and gold edge card connectors. Internal drive systems include an interface card and a half-height hard drive that replaces the top disk drive in both size and power consumption. External units include an interface card, case, power supply and hard drive unit. All units require a memory board with DMA such as the TanPak™ multifunction card, or a Tandy Board, Part #25-1004.

Hard Disk System Features:

- All sizes rated after formatting.
- One secondary drive may be added.
- Error checking and correcting controller.
- Software drivers included.
- Buffered seek drives for improved access time.
- Built in power up diagnostics.
- Plated media for long disk life.
- Gold connectors used throughout.
- 1 year warranty.
- Boots directly from Hard Drive.
- Uses Tandy 1000 MS DOS.

10 Megabyte Internal	**$ 549.**
20 Megabyte Internal	**$ 749.**
42 Megabyte Internal	**$1349.**
10 Megabyte External	**$ 699.**
20 Megabyte External	**$ 899.**
42 Megabyte External	**$1495.**

Ordering Information

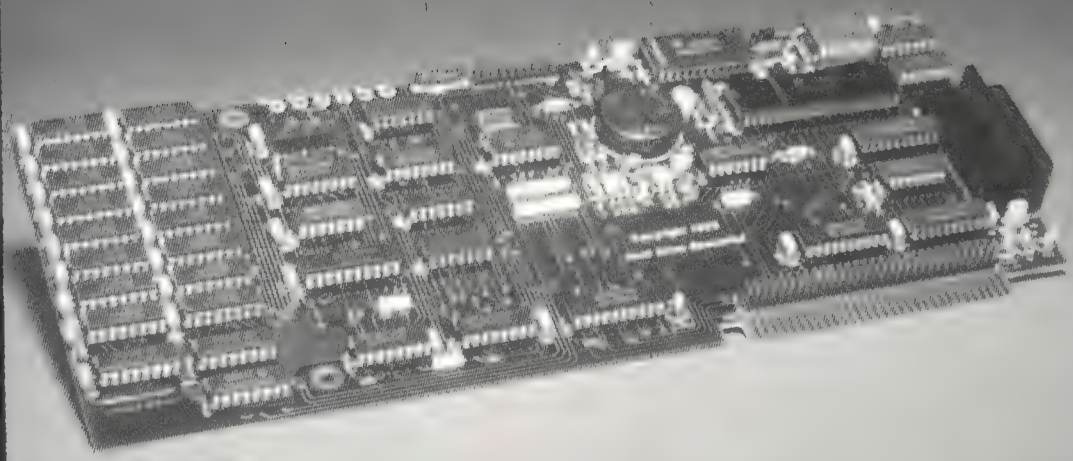
Use our WATS line to place your order via Visa, MasterCard, or Wire Transfer. Or, mail your payment directly to us. Any non-certified funds will be held until proper clearance is made. COD orders are accepted as well as purchase orders from government agencies. UPS ground is our standard means of shipping unless otherwise specified. Shipping costs are available upon request. Tandy and Radio Shack are trademarks of the Tandy Corporation. TanPak, is a trademark of Hard Drive Specialist, a division of Compukit Corporation.

Hard Drive Specialist

16208 Hickory Knoll
Houston, Texas 77059
1-713-480-6000

Order Line
1-800-231-6671

The Ultimate Expansion for the Model 1000!
TanPak ™ by Hard Drive Specialist



7 Functions Include:
DMA
Serial Port
Memory up to 512K
Clock/Calendar
Printer Spooler
Memory Disk
Expansion Port for Future Options

Features Include:
Gold Edge Cards
Pre-Tested and Burned In
Full Documentation
Supporting Software
One Year Warranty
Expandable to 512K

PCM CERTIFICATION SEAL OF QUALITY

TanPak ™

The TanPak ™ expansion board has been designed to allow expansion beyond the scope of the standard Model 1000. Seven of the most needed functions/features have been combined into one package using only one expansion slot. Your remaining spaces are left free for future expansion needs.

Seven Function/Features on One Board

512K Memory Expansion

Socketed and expandable to 512K. This is done by two banks of memory using either two 64K increments (128K), or 256K increments (256K or 512K). This allows a total of 640K in the Tandy 1000.

Serial Port

Using the same configuration as the Model 1000 port you are assured of complete compatibility as well as being able to configure it as COM1 or COM2.

Clock-Calendar

Quartz-controlled for a high degree of accuracy, featuring a battery backup.

DMA

The DMA (Direct Memory Access) is used on the Model 1000's first memory card. It increases memory speed and is a must for hard drive operation.

Printer Spooler

Use part of your TanPak ™ memory as a printer buffer. Choose the amount of buffer space you need and stop waiting on your printer.

Memory Disk

Use part of your TanPak ™ memory as a RAM drive. With a solid state drive you can store, retrieve, and sort data quickly and easily.

Expansion System

The TanPak ™ was designed with an expansion port that will allow upgrading with additional features when they become available. Some of the possible features are: a second parallel port, a second serial port, mouse, hard drive port, bubble memory, A/D, and D/A as well as many others.

TanPak ™ 0K . . . . . $329.

Call for pricing on memory modules.

TanPak ™ Secondary

If you already have a Model 1000 memory board and do not wish to replace it, the TanPak ™ Secondary is for you. It features Memory up to 256K, Clock/Calendar, Serial Port, Printer Spooler, and Printer Buffer.

TanPak ™ Secondary 0K . . . . . $249.

TanTel ™

Internal 300/1200 baud Modem for the Tandy 1000. Supported by the DeskMate software package.

TanTel ™ . . . . . $249.

Ordering Information

Use our W.A.T.S. line to place your order via Visa, MasterCard, or Wire Transfer. Or mail your payment directly to us. Any non-certified funds will be held until proper clearance is made. COD orders are accepted as well as purchase orders from government agencies. UPS ground is our standard means of shipping unless otherwise specified. Shipping costs are available upon request. Tandy and Radio Shack are trademarks of the Tandy Corporation. TanPak, and TanTel are trademarks of Hard Drive Specialist, a division of Compukit Corporation.

Hard Drive Specialist
16208 Hickory Knoll
Houston, Texas 77059
1-713-480-6000
Order Line
1-800-231-6671

Mills, Bob. "Metric calculator." *(1986, June)* 73• — Converting measurements with PASCAL.
Mills, Robert C. "Galactic math war." *(1986, April)* 10• — Math tutor game for the Model 1000.
Murray, Dennis ; and Ory, Horace. "Efficient allocation of hard disk storage." *(1985, December)* 55
Nickols, Kevin. "Delphi bureau: 'The' gathering place for Tandy computer users." *(1986, March)* 30
Nickols, Kevin. "Delphi bureau." *(1986, February)* 38 — Latest news from Delphi.
Nickols, Kevin. "Delphi bureau." *(1986, April)* 54
Nickols, Kevin. "Delphi bureau." *(1986, May)* 108 — Uploading of files.
Nickols, Kevin. "Uploading files to Delphi." *(1986, June)* 97
Oppedahl, Carl. "Emergency power for your Tandy 600." *(1986, February)* 52
Oppedahl, Carl. "Manipulating RAM files in the Tandy 200." *(1985, August)* 57
Oppedahl, Carl. "Wrist terminal." *(1986, April)* 33 — Close look at the Seiko wrist terminal.
Petersen, Marty. "The power of Open Access." *(1985, September)* 68 — A close look at this integrated program.
Preble, Laurence D. "If X equals zero the X equals ten trillion." *(1985, September)* 39 — Discussion of a 'bug' in the Model 200 BASIC.
Qualls, Bill. "A better input processor." *(1985, August)* 49 — Prevent data entry errors.
Ramella, Richard. "Add search and replace power to your portable's text-handling functions." *(1985, September)* 25★
Ramella, Richard. "Bit parade." *(1985, December)* 30 — Compose melodies.
Ramella, Richard. "BOA: The main squeeze for text files." *(1986, April)* 36★• — How to compact text files.
Ramella, Richard. "Sardine." *(1986, March)* 22★ — Pack programs to conserve space.
Ray, J. D. "Landlord's helper." *(1986, May)* 10• — Rental property management.
Ray, James. "Inflation tracker." *(1986, March)* 49•
Sanders, Wayne. "The gallery." *(1985, July)* 48 — Graphics demo.
Sanders, Wayne. "The gallery." *(1985, August)* 28
Sanders, Wayne. "The gallery." *(1985, September)* 66
Sanders, Wayne. "The gallery." *(1985, October)* 57
Sanders, Wayne. "The gallery." *(1985, November)* 60
Sanders, Wayne. "The gallery." *(1985, December)* 52
Sanders, Wayne. "The gallery." *(1986, January)* 66•
Sanders, Wayne. "The gallery." *(1986, February)* 49•
Sanders, Wayne. "The gallery." *(1986, March)* 44•
Sanders, Wayne. "The gallery." *(1986, April)* 20•
Sanders, Wayne. "The wind chill factor." *(1986, February)* 36★•
Scerbo, Fred. "Portable multiple choice test generator." *(1985, August)* 66★
Shewchuk, John B. "Star Trek." *(1986, June)* 50• — Game.
Stajduhar, Jerry. "The music machine." *(1986, May)* 33• — Create, modify and save songs.
Teague, John F. "Screen to memory to screen." *(1985, September)* 59★ — Store and retrieve graphics.
White, Richard A. "BASIC control structures." *(1986, June)* 46 — Tutorial on FOR—TO—NEXT.
White, Richard A. "A brief survey of basic spreadsheet functions." *(1985, November)* 36
White, Richard A. "The car shopper's spreadsheet." *(1985, July)* 66 — Correction, August 1985, p. 31.
White, Richard A. "Different ways to store program data." *(1986, May)* 77
White, Richard A. "Form 1040 in a spreadsheet." *(1986, March)* 82
White, Richard A. "Let a spreadsheet dispel the gloom of form creation." *(1985, August)* 31
White, Richard A. "Marrying BASIC and a spreadsheet." *(1985, October)* 68
White, Richard A. "Preparing for the tax collector." *(1985, September)* 75 — Use the spreadsheet for tax time.
White, Richard A. "Spreading on more spreadsheet functions." *(1985, December)* 17
White, Richard A. "Using the spreadsheet as a scheduling tool." *(1986, January)* 63
White, Richard A. "Welcome to BASIC." *(1986, April)* 100 — Tutorial on BASIC.

BAR CODE LISTINGS

Balonis, Ronald F. "A text analyzer for the Model 100/200." *(1986, March)* 95★
Bishop, Dan. "Color code combo." *(1986, June)* 100★• — Deductive reasoning game.
Chatham, Art. "April fool!" *(1986, April)* 102★• — Appears if your files have vanished.
Cornman, John ; and Cornman, Aileen "Easy errors." *(1986, February)* 23★• — Better error messages.
Halcomb, Jay. "A SCHEDL enhancement." *(1985, July)* 29★
Henning, Michael R. "Model 100 machine code made easy." *(1986, June)* 86★• — Portable machine language programming.
Himowitz, Michael J. "Big print." *(1986, April)* 58★• — Print in large letters.
Ingle, Frank H. "Coping with cassettes." *(1986, May)* 49★•
Ireland, Nathaniel F. "Portable trip companion." *(1985, October)* 38★
Ireland, Nathaniel F. "Your 100 takes a licking and keeps on ticking." *(1986, May)* 30★• — Indicate elapsed time in hundreths of a second.
Krueger, Roger F. "The fantastic label machine." *(1985, November)* 13★ — Make mailing labels.
Larrison, John. "Super editor." *(1986, April)* 23★• — Machine code editor.
McDowell, Linwood. "Easy keys." *(1985, December)* 32★ — Creative use of function keys.
Miller, Chris. "Portable protection." *(1986, January)* 79★ — Hide your files on the portable. Correction, March 1986, p. 9.
Ramella, Richard. "Add search and replace power to your portable's text-handling functions." *(1985, September)* 25★
Ramella, Richard. "BOA: The main squeeze for text files." *(1986, April)* 36★• — How to compact text files.
Ramella, Richard. "Sardine." *(1986, March)* 22★ — Pack programs to conserve space.
Sanders, Wayne. "The wind chill factor." *(1986, February)* 36★•
Scerbo, Fred. "Portable multiple choice test generator." *(1985, August)* 66★
Teague, John F. "Screen to memory to screen." *(1985, September)* 59★ — Store and retrieve graphics.

PCM ON DISK

ADDRESS Ballard, Bobby. "Return address label maker." *(1986, May)* 80•
BARDEN Barden, William, Jr. "Interfacing with the Tandy 1000's joystick ports." *(1986, January)* 34•
BARDEN Barden, William, Jr. "What are all those strange MS-DOS commands, anyway?" *(1986, March)* 70•
BARDEN Barden, William, Jr. "What are all those strange MS-DOS commands, anyway part 2." *(1986, April)* 40•
BARDEN Barden, William, Jr. "How to make cents of your MS-DOS system." *(1986, May)* 96• — Methods of displaying numbers.
BARDEN Barden, William, Jr. "What a bunch of characters!" *(1986, June)* 10• — Create new character sets on the screen.
BEAR Hyre, Leonard. "The land of the cave bear." *(1986, March)* 10• — Adventure game.
BIGPRINT Himowitz, Michael J. "Big print." *(1986, April)* 58★• — Print in large letters.
BOA Ramella, Richard. "BOA: The main squeeze for text files." *(1986, April)* 36★• — How to compact text files.
CLOCK Cornman, Aileen ; and Cornman, John. "It's about time—Multi-clock." *(1985, October)* 59• — Upgrade the Model 100 timekeeping. Note: Contained in April 1986 PCM ON DISK.
COMBO Bishop, Dan. "Color code combo." *(1986, June)* 100★• — Deductive reasoning game.
COPING Ingle, Frank H. "Coping with cassettes." *(1986, May)* 49★•
DESK Ballard, Bobby. "The integrated desk part 7: Turbo powered DeskMate." *(1986, May)* 87•
DOSBOOT Harrell, John B., III. "Superfast displays." *(1986, January)* 67•
DOSBOOT Harrell, John B., III. "A batch of new ideas." *(1986, March)* 77• — Tutorial on MENU.-COM.
DOSBOOT Harrell, John B., III. "Security's the name of the game." *(1986, April)* 75• — Password protection.
DOSBOOT Harrell, John B., III. "What makes a compatible." *(1986, May)* 41•
EASY Henning, Michael R. "Model 100 machine code made easy." *(1986, June)* 86★• — Portable machine language programming.
ELECTRIC Hyre, Leonard. "The electric blackboard." *(1986, February)* 10• — Mathematics exercises.
ERRORS Cornman, John ; and Cornman, Aileen "Easy errors." *(1986, February)* 23★• — Better error messages.
FILTERS Kemp, H. W. "Fun with filters." *(1986, January)* 48• — Graphics demos. Corrections, March 1986, p. 105.
FOOL Chatham, Art. "April fool!" *(1986, April)* 102★• — Appears if your files have vanished.
GALLERY Sanders, Wayne. "The gallery." *(1986, January)* 66•
GALLERY Sanders, Wayne. "The gallery." *(1986, February)* 49•
GALLERY Sanders, Wayne. "The gallery." *(1986, March)* 44•
GALLERY Sanders, Wayne. "The gallery." *(1986, April)* 20•
HOUSE Baade, Eisner H. "Household inventory." *(1986, March)* 36•
INFLATE Ray, James. "Inflation tracker." *(1986, March)* 49•
IRA Hyre, Leonard. "IRA projection." *(1986, March)* 46•
KEYS McCormick, John. "Hitting the right keys." *(1986, June)* 98• — Improve numeric keypad skills.
LANDLORD Ray, J. D. "Landlord's helper." *(1986, May)* 10• — Rental property management.
LOAN Hyre, Leonard. "The loan officer." *(1986, May)* 57• — Amortization schedules.
MATHWAR Mills, Robert C. "Galactic math war." *(1986, April)* 10• — Math tutor game for the Model 1000.
METRIC Mills, Bob. "Metric calculator." *(1986, June)* 73• — Converting measurements with PASCAL.
MUSIC Stajduhar, Jerry. "The music machine." *(1986, May)* 33• — Create, modify and save songs.
PHONE Hyre, Leonard. "PC phone booth." *(1986, June)* 33 — Mailing list, phone number directory.
PSU DuBois, Marshall K. "A handy little printer setup utility." *(1986, February)* 78•
RANDOM Bruey, Alfred J. "Using random files." *(1986, April)* 25•
RICKETY Jackson, Barb ; and Jackson, Joan. "Rickety downs." *(1986, January)* 31• — Horse racing game.
SCRNDUMP Lee, Peter T. "Graphics screen dump." *(1986, March)* 100• — Correction, May 1986, p. 109.
SUBCITY Covington, Robert D. "DB11: A towering database system." *(1986, February)* 57•
SUBCITY Covington, Robert D. "Compiling BASIC programs." *(1986, March)* 56• — Discussion of the GW-BASIC compiler, and updates to DB-11 database. File SUBCITY in April PCM ON DISK contains further updates.
SUPER Larrison, John. "Super editor." *(1986, April)* 23★• — Machine code editor.
TICKING Ireland, Nathaniel F. "Your 100 takes a licking and keeps on ticking." *(1986, May)* 30★• — Indicate elapsed time in hundreths of a second.
TREK Shewchuk, John B. "Star Trek." *(1986, June)* 50• — Game
VIDEO Crockett, Davy. "Video character editor." *(1986, May)* 66• — Change display on screen in Model 2000.
WINDCHILL Sanders, Wayne. "The wind chill factor." *(1986, February)* 36★•

The Return of the Menu

By John B. Harrell, III

PCM Contributing Editor

I previously talked about using multiple levels with the *Menu* program and stated that the solution presented was a "work-around" the limitations. The techniques for using interactive batch files are useful, but this is less than the optimal solution for your menu needs.

My original *Menu* has some other limitations, too. I had two different versions depending on your specific machine. I had to write a special program for installing *Menu*. There was no really easy way to exit from *Menu*. It was written in assembly language and, therefore, was harder to maintain.

Menu always used the command processor from the disk currently in use when it was loaded. This feature seems relatively insignificant, *but* you can install a RAM disk and copy COMMAND .COM to it. Then, whenever the command processor needs loading, the system is able to perform this task much faster than loading it from disk.

John B. Harrell, III has written for microcomputer magazines for three years. He holds a bachelor's degree in computer science and is a software technical expert for Navy electronic support measures systems.

Super *Menu*

Listing 1 contains the C source listing for a revised version of *Menu* that runs on any Tandy MS-DOS or IBM PC compatible, including the Tandy 2000 computer without change. In the following paragraphs, I will explain the differences between this version and the previous version.

I wrote the revised version of *Menu* in C so I could devote my time to some nicer features without having to worry over the details of writing the assembly language code. I also wanted to introduce you to another popular high-level language as an alternative to

Turbo PASCAL. Using C does have a penalty in increased code size and slower execution speed, but the difference is not enough to worry about, considering the advantages.

I used Manx's *Aztec-C86* compiler for this project. You can obtain this compiler in a number of configurations and it is an excellent compiler for any program development. Aztec produces exceptionally small compiled programs and runs as fast (and faster in some cases) than Microsoft's Version 3. If you are considering C for a new language to learn, this is an exceptional product and Manx also has a C learning system and interpreter that is heavily related to their compiler.

Because each compiler implements non-standard library features in different ways, I have included a description in the comments of Listing 1, illustrating the Aztec function name and its usage. Most current C compilers implement these same functions, only with different names and calling sequences. Portability to another compiler should be easy.

What's New with *Menu* Version 2

I never liked the klutzy installation method used with the first version of *Menu*. I would start to change the installation on someone's computer and never had MENUINST.BAS around when I needed it. I often had to attempt a modification with Debug — not the most desirable circumstances. The new *Menu* uses a set-up file consisting of plain ASCII text created with Edlin or any word processor capable of writing a DOS text file. Later, I'll describe how to build this file.

The previous *Menu* installation would have supported multiple levels of menus if you wanted to install a version for each level and could depend on a CTRL-BREAK (or CTRL-C) to terminate the secondary levels. I did not like this idea, mainly because the concepts were too difficult to explain to the person who just opened his computer box. *Menu* was meant to relieve stress and anguish, not create it!

The new version expects to find the file SETUP.MNU in the current directory for initialization. You can also execute *Menu* with the complete pathname for the set-up file on the command line. For example, MENU C:\DOS\UTIL\SE COND.SET executes

In addition to a price you can afford more easily:

[1]A buy-back policy*, to insure you against a good idea that turned out to be a mistake or even if you simply "changed your mind". [2]An exchange, for a new one, in the rare event that yours should prove to be a "lemon". [3]Credit card purchasing convenience for speed and safety plus [4]an additional discount for cash-with-order buyers and [5]a CALL-FREE number so you can get our prices free (except Texas). [6]A price quote which has NO ROAD-FREIGHT charges added later, we pay the freight and [7]fast

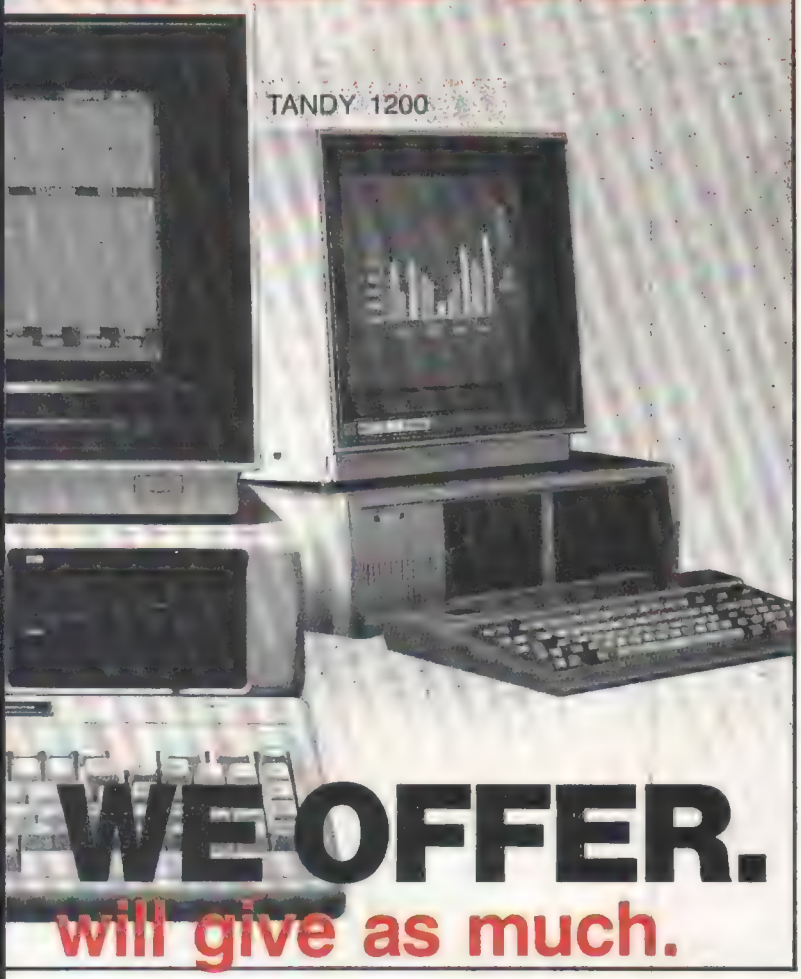
* For a small restocking fee.

CUSTOMER SERVICE/QUESTIONS ABOUT YOUR ORDER and in TEXAS 1-817-573-4111
(9 am-5 pm TEXAS TIME MONDAY-FRIDAY)

Fort Worth Computers
377 Plaza
Granbury, Texas 76048

CALLFREE NUMBER:

COMPUTERS and PRINTERS etc.



WE OFFER.
will give as much.

shipping, usually next business day. [8]A same-day "rush service" for late satisfaction and [9]the assurance that repeat-customers exceed new ones means great satisfaction. [10]References given upon request, to prove our bona-fides.

Tandy computers are created equal. . .all retailers are not.

Did you get "the rest of the story" before you placed your order?
(. . .good thinking. . .)

VISA AMERICAN EXPRESS MasterCard

FORT WORTH COMPUTERS

(WE ARE SERIOUS ABOUT SAVING YOU MONEY)

Located 30 miles from Ft. Worth

For Latest Prices (1-800-) 433-S-A-V-E

Menu and looks for the initialization file SECOND.SET on the hard disk Drive C in the UTIL subdirectory of the DOS subdirectory.

The set-up file contains the screen colors, time delay, title and menu options. Up to 24 options can be set in the menu and each option can specify a complete program name or batch file, including parameters. I got tired of the cryptic batch files (A, B, C, etc.) after a while with the older version. If you use batch files for the menu options (you really should use them to manage the subdirectories), they can now be given descriptive names so you will be able to remember them later. Floppy disk users will appreciate this the most as it totally frees them from having to use batch files without renaming their program files.

This version solves another big gripe that floppy disk drive users (and some hard disk users, too) have expressed. The original version was capable of formatting a floppy disk in Drive A only. *Menu* now gives you the choice of Drive A or B.

Menu does not allow you to exit to the DOS command level. Users most often wanted only to execute a single command and return to the menu. *Menu* now provides a full "DOS window," which executes the single DOS command typed. Once *Menu* executes the command, the screen is left intact for you to read and a prompt to press any key is written on the bottom screen line.

There is an "escape mechanism" — press ALT-1 and you exit the current *Menu* shell. This is also the method used to exit from a secondary menu level back to the primary level. You can also use the DOS window to execute the command processor. Type COMMAND and press ENTER — you will be at the DOS command level until you type EXIT.

The final improvement provides the capability of loading the command processor from a RAM disk or any alternative other than the boot disk. When MS-DOS initializes, it creates the environment variable COMSPEC with the full pathname containing the location of the DOS command processor. Type the DOS command SET after booting the system and see what you have installed. The *Aztec-C* system function automatically searches the current program environment and uses the command processor from the

location specified.

Install your RAM disk driver per normal instructions. You only need about 16K to 27K of disk space depending on your DOS version of COMMAND .COM — look at a directory listing containing COMMAND.COM to find out. Next, place these two commands in your AUTOEXEC.BAT file just prior to executing the new version of *Menu*:

```
COPY COMMAND.COM drive :
SET COMSPEC=drive: \COMMAND.COM
```

Drive is the drive designation for your RAM disk.

Any time *Menu* executes one of the options, it loads the DOS command processor from the location specified by COMSPEC. This is a lot quicker than loading it from disk.

Initial *Menu* Setup

Load your favorite DOS text processor (Edlin) or any word processor capable of generating a DOS text file. The first five entries are numeric and contain the colors and time delay for menu. The first four are the screen colors and are in hexadecimal. For the Tandy 1000, 1200 and 3000, these colors reflect the background and foreground of each character and use the same numbering scheme as presented in the original article (November 1985, Page 71). For the Tandy 2000, each represents a single number, also as previously explained.

Color 1 is the pop-up window containing the formatting message and part of the DOS window. Color 2 is the second format message and the main color for the DOS window. Color 3 is the menu body and text. Color 4 is for the title lines at the top and bottom of the menu. For the Tandy 2000, colors 1 and 2 specify the highlight background and foreground colors, and colors 3 and 4 specify the normal background and foreground colors.

The fifth number is in decimal and specifies the number of seconds to delay prior to protecting the screen with the flashing prompt. These numbers must be separated by a "white space" character: a blank, tab or line separator. I put each number on a separate line for clarity.

The next line of the set-up file contains the title for your menu and can be any character string up to 78 characters long. It is on a single line by itself — just type the title text. Following the title, each menu entry is listed with its corresponding command.

The menu text for each item may be any text string up to 35 characters in length. Each menu item must be followed by a command string that is permanently assigned to that item. Menu items are positional. Where they appear in the set-up file establishes the letter assigned to that option. If you are going to skip over several menu option letters, that's fine — just remember to leave two blank lines for each item skipped.

Up to 24 distinct command options can be specified for each menu. You may stop wherever you want to — *Menu* automatically fills out the remainder of the list with blank options. If an error is detected in the menu options, *Menu* displays the last text item read and terminates in error.

Write the set-up file out to disk and you are all set to go. One last thing, you must also rename FORMAT.COM to FMT.COM. I still believe in this modest form of disk protection enough to use it in the revised *Menu*.

Conclusion

Menu contains all of the C code for rapidly displaying text to the video screen. See my "Superfast Displays" article (January 1986). I modified this code to recognize the difference between a Tandy 2000 and an IBM-compatible computer. With this modification, I can use the same identical code for all machines.

I think the C source code is fairly well-documented and easy to read as it is. If you try to convert the code to your compiler and use it, C is case-sensitive — the source must be typed as it appears. Please write with any questions or send them to me via Easyplex mail on CompuServe — my user ID is 73016,1326. □

Reductions of actual signs

#1 6.9% FINANCING

MAKE SIGNS with the BANNER MACHINE

(PC & compatibles)

REGISTER HERE! A-F

See review in PCM 6/86 p. 105

TODAY'S SPECIALS

THINGS	$ 4.95
WIDGETS	$ 6.95
DELUXE WIDGETS	$17.50
SOOPER DOOPER	$19.95

Create useful signs for:
- Business Presentations
- Employee Announcements
- Product Information (for convention booths)

Professional quality signs up to 14" tall by any length. Mix letter sizes and fonts on the same sign in one pass. 10 letter styles in 8 sizes from 3/4" to 6 1/2" high included. Also includes graphic symbols, textured background shades, 30 borders. Easy to use. Signs can be saved on disk for use at a later date.

SPECIAL OFFER!

The BANNER MACHINE *(with 4 fonts, graphic symbols, 20 shades & 12 borders!* $79.95

Font Disk *(6 additional fonts & 18 borders)* $79.95

Complete package $149.95

800 762-5645

Cardinal Software 14840 Build America Drive, Woodbridge, VA 22191 Info: (703) 491-6494

The listing:

```
/*****************************************************************************
 *
 *                          MENU Utility Shell
 *
 *                    Written by: LCDR J. B. Harrell
 *
 *****************************************************************************/

/* :bk=Ø */

/*****************************************************************************
 *
 *
 *      Documentation of Aztec-C86 compiler library specific functions
 *
 *
 *  scr_call(ax,bx,cx,dx) is an Aztec function to perform a Video Control
 *  Interrupt (INT 1ØH) with the AX, BX, CX, and DX registers set to the
 *  values passed as parameters to the function call.
 *
 *  scr_poll() is an integer function which interrogates the keyboard
 *  using INT 16H and returns the value of the key pressed, if any, or
 *  -1 for no key, -2 for Ctrl-Break.  Extended keys such as function keys
 *  return their scan code plus 128.
 *
 *  scr_getc() is an integer function which interrogates the keyboard
 *  using INT 16H and returns the value of the key pressed.  It WAITS for
 *  a key to be pressed.  scr_getc() returns -2 for Ctrl-Break and handles
 *  function keys as above for scr_poll().
 *
 *  scr_puts() and scr_printf() are Aztec functions which perform the
 *  equivalent of the normal C library puts() and printf() functions using
 *  BIOS INT 1ØH calls.  These functions will also use the currently defined
 *  screen attribute to display text in colors.
 *
 *  scr_loc(&row,&col) uses BIOS INT 1ØH to retrieve the current cursor
 *  location and set the parameter addresses specified.
 *
 *  scr_curs(row,col) uses BIOS INT 1ØH to set the system cursor to the
 *  location specified by "row,col".
 *
 *  sysint(int,&iregs,&oregs) sets the CPU registers to the values specified
 *  in the structure "iregs" and executes software interrupt "int".  The
 *  returned register values are stored in "oreg".
 *
 *  system(str) executes the command string passed as the parameter.  The DOS
 *  command processor (COMMAND.COM) is loaded from the file identified by the
 *  environment variable "COMSPEC".  Null strings will not be executed.
 *
 *  outportb(port,byte) outputs the byte specified to the CPU port.
 *
 *  peekb(offset,segment) is an integer function returning the byte stored in
 *  location specified by "segment:offset".
 *
 *  pokew(offset,segment,word) stores the word specified at the word location
 *  specified by "segment:offset".
 *
 *****************************************************************************/

#include <stdio.h>

#define CLS         scr_call(Øx6ØØ,Øx7ØØ,Ø,Øx184F)     /* B

IOS call */
#define CURSOR_ON   scr_call(Øx1ØØ,Ø,on_value,Ø)        /* BIOS call

 */
#define CURSOR_OFF  scr_call(Øx1ØØ,Ø,off_value,Ø)       /* BIOS call

 */
```

```
#define TRUE        1
#define FALSE       Ø

#define TANDY2ØØØ   1
#define IBMMONO     2
#define IBMCOLOR    3

#define ESCAPECHR   Øxf8            /* currently defined as the ALT-1 key   */
                                    /* using Aztec's notation of 128 (8ØH)  */
                                    /* plus the scan code of the key        */

struct regs {                       /* used for "sysint" communications     */
    int AX, BX, CX, DX, SI, DI, DS,  ES;
};

/*
 *      The constants listed below are values used by the Aztec-C86 run-time
 *      system to control allocation of the stack and heap.  You are also
 *      able to force the compiled program to generate the stack below the
 *      heap, decreasing the size of the execuatble code.
 *
 */

int _STKSIZ =  128;     /* stack size in paragraphs                          */
int _HEAPSIZ = 128;     /* heap size in paragraphs                           */
int _STKLOW =  1;       /* Force stack below heap (small memory model only) */

/*
 *
 *      These constants are used by the video display routines used in MENU
 *
 */

int SMODE   = 3;        /* Assumed video screen mode, reset by init_video() */
int Video_Mem;          /* Segment address for video display RAM            */
int vflag;              /* State variable containing current video mode     */
int attrib;             /* Current character attribute value                */
int normal;             /* Actual attribute for "normal video"              */
int revrs;              /* Actual attribute for "revers video"              */
int hilite;             /* Actual attribute for "highlight video"           */
int rhilite;            /* Actual attribute for "reverse highlight video"   */
int on_value, off_value;/* Cursor control values                            */

int nor  = 1;           /* Local                                            */
int rev  = 2;           /*       attribute                                  */
int hnor = 3;           /*                 selection                        */
int hrev = 4;           /*                           values                 */

/*
 *      The following data items contain the information read from the menu
 *      setup file (either SETUP.MNU or the file name specified on the DOS
 *      command line.
 *
 */

int NORMAL;             /* Color attribute                                  */
int REVRS;              /*                values read                       */
int HILITE;             /*                            file                  */
int RHILITE;            /*                                 SETUP.MNU        */
int TDELAY;             /* Time delay (seconds) before blanking screen      */
char title[6Ø];         /* Title line text informaiton                      */
char menuitems[26][38]; /* Menu entry text (one for each menu line)         */
char menucommand[24][5Ø];/* Menu command text (one for each menu entry      */
                        /* with the exception of the last two items which   */
                        /* are reserved for formatting disks and the DOS    */
                        /* command window                                   */

int ch, ch2, screenflag;
long starttime, lasttime, thistime, gettime();
char cmd[8Ø];
char setupfile[63];
char *prompt = "Press a letter key for your selection";
char *menuver = "MENU Version 2.3";
```

```
/*****************************************************************************
 *
 *                      Main program function
 *
 *****************************************************************************/

main(argc,argv)
int argc;
char *argv[];
{
    if ( argc == 1 )    {
        strcpy(setupfile,"SETUP.MNU");
    }
    else    {
        if ( argc == 2 )    {
            strcpy(setupfile,*++argv);
        }
        else    {
            scr_puts("\nMENU parameter error - only filename allowed!");
            exit(1);
        }
    }

    init_menu();
    screenflag = FALSE;

    do  {                           /* main Menu control loop */

        starttime = gettime();      /* Initialize blanking delay */
        lasttime = 0L;

        if ( !screenflag )  {       /* Turn on video if it is off   */
            init_video();
        drawmenu();
    }

    while ( scr_poll() == -1 ) {    /* read keyboard and update screen */
        if (((thistime=gettime()) > starttime+(long)TDELAY) && screenflag)
            dead_screen();
        if ( lasttime != thistime ) {
            lasttime = thistime;
            if ( screenflag )   {
                header(22,0,79,cmd,hrev);
            }
            else    {
                dead_screen();
            }
        }
    }

    if ( !screenflag )  {
        init_video();
        drawmenu();
    }

    ch = toupper(scr_getc());       /* decipher command key pressed */
    if ( (ch >= 'A') && (ch <= 'X') )   {
        doscommand(menucommand[ch-'A']);
    }
    else    {
        switch ( ch )   {           /* was not A through X  */
            case 'Y':
                formatdisk();
                break;
            case 'Z':
                get_dos_command();
                break;
        }
    }
}   while ( ch != ESCAPECHR );

cls();                              /* restore video to normal and exit */
scr_call(SMODE);
```

```
    CURSOR_ON;
}

/*****************************************************************************
 *
 *      init_menu()
 *          Initialize the menu constants by reading the data from the
 *          appropriate setup file.
 *
 *****************************************************************************/

init_menu()
{
    FILE *setup, *fopen();
    int i;
    char *index(), *cp;
    char buffer[8Ø];
    char *fgets();

    cls();
scr_puts(menuver);
scr_puts("Written by John B. Harrell");

strcpy(buffer,"A. ");                   /* Initialize initial letters of    */
for ( i = Ø; i < 26; i++ )  {           /* MENU entries, "A" through "Z"    */
    strcpy(menuitems[i],buffer);
    buffer[Ø]++;
}

if ( (setup = fopen(setupfile,"r")) == NULL )   {
    scr_printf("Setup file <%s> not found!",setupfile);
    exit(1);
}

if ( fscanf(setup,"%x %x %x %x %d",
        &NORMAL,&REVRS,&HILITE,&RHILITE,&TDELAY) != 5 ) {
    scr_puts("Error in setup file numeric color input:");
            scr_puts("  Required input is four numbers in hexidecimal");
            scr_puts("  representing the screen colors followed by one");
            scr_puts("  number in decimal representing the time delay.");
            scr_puts("  These numbers must be separated by blanks, tabs");
            scr_puts("  or on a separate line.");
    exit(1);
}

if ( getc(setup) == EOF )   {
    scr_puts("Error in setup file text options at title line");
    exit(1);
}

/*
 *      In the following code segment, MENU reads the setup file and
 *      extracts the text entries and commands.  Note that fgets() will
 *      return a string terminated with '\n' and the program uses
 *      index() to locate this character and replace it with '\Ø'
 *
 */

if ( fgets(buffer,78,setup) == NULL )   {
    scr_puts("Error reading menu title");
    exit(1);
}
cp = index(buffer,'\n');
*cp = '\Ø';
strcpy(title,buffer);

for ( i = Ø; i < 24; i++ )  {
    /*
     *  Up to 24 menu entries may be present.  An end of file [fgets()
     *  returns a NULL value] will be received if less are present and
     *  this terminates the loop.  If a menu text entry is present, the
     *  command text must also be present or this is signaled as an error
     *  condition.  Menu text entries may be 35 characters long and
     *  command text may be up to 5Ø characters.
```

TURBO-M™

Check Out Our Systems — High **QUALITY** ***TURBO*** motherboard at the **LOWEST** prices.
Our hard drive systems have **EXTRA** megabytes of ***FREE*** storage — ***CALL***
(more megabytes for less megabucks)
Our monitors are high resolution TTL — *They even have their own tilt and swivel base* ***FREE***
Powered by a heavy duty power supply, and offering ***TRUE*** — **8Mhz turbo speed**
ONE FULL YEAR WARRANTY

COMPARE OUR SYSTEMS — TURBO M is the COMPETITION

SYSTEM F
FLOPPY SYSTEM
$899.95!

System H.D.
HARD DRIVE SYSTEM
22 MEGABYTE SYSTEM
$1,399.95!

TURBO M PC COMPLETE SYSTEM

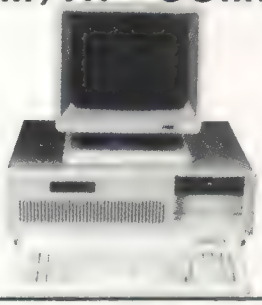
Dual Speed CPU — 4.77 AND 8Mhz TURBO
IBM COMPATIBLE

640 K memory
8 slot motherboard
150 Watt Power Supply
Monochrome TTL Card
Monochrome Display High Res.
Tilt and Swivel Base Monitor
Serial port (2nd port opt.)
2 parallel ports
Clock calendar
Game Port
Keyboard 5151 style
Ram spooler
Printer spooler
DOS 2.11
G.W. Basic

Toll Free Ordering 1-800-343-8841

TURBO M/AT—COMPLETE SYSTEMS STARTING AT $2,359.95 !!!

ONE FULL YEAR WARRANTY



Comes Complete with 80286 processor
6 and 8 Mhz Switch Selectable
8 Slot Motherboard
192 Watt Power Supply
1.2 Megabyte Floppy Drive
1 Full Megabyte of Memory on Motherboard
2 Parallel Ports — 2 Serial Ports
Clock Calendar
Hercules Compatible Monochrome Card
Monochrome Display-High Resolution
Tilt and Swivel Base
AT Keyboard

ADD-IN BOARDS TO HELP YOU BUILD YOUR SYSTEM

TURBO-M Motherboad Dual Speed **$199.95**	Monochrome Graphics Card with Printer Port **$79.95**	Floppy I/O Card Controls Two Floppy Drives, Serial Port Parallel Port Clock Calendar Game Port **$99.95**	Winchester Hard Drive Controller **$134.95**	10 Megabyte and 20 Megabyte Drives Complete System Drive Controller Cables and easy to follow instructions **$349.95 — $459.95**		Color Graphics Card with Printer Port **$94.95**
150 Watt Power Supply **$79.95**	Multi Function Card with Space for 384K of Memory, Serial Port, Parallel Port Clock Calendar, Game Port **$79.95**	5151 Compatible Keyboard with Separate Cursor and Number Pads **$89.95**	Monochrome Monitor with High Resolution Tilt/Swivel Base **$159.95**	Floppy Drive Card Control up to Four Drives **$39.95**	P.C.DOS 3.1 **$80.00**	Flip Top Cabinet with Speaker **$49.95**

MEGADISK™ HARD DRIVES

Our Hard Drive Systems Are Compatible With
IBM PC and AT, HEATH/ZENITH, COMPAQ, COLUMBIA, PC CLONES, TAVA, EAGLE
TRS/80 Models I / III / IV / 4D / 4P / MAX 80 / COLOR COMPUTER, Tandy 1000

Systems come complete with drive, controller, cables, hardware and easy to follow instructions.

IBM ETC
6 megabytes .... starts at $229.95
11 megabytes .... starts at $349.95
22 megabytes .... starts at $459.95

REMOVABLE MEDIA SPECIAL
5 Megabyte Cartridge Hard Drive Now
$399.95

Systems come complete with one free software driver for TRSDOS 6, LDOS 5.1.x, DOSPLUS 3.4/4, or NEWDOS 80.CP/M is available.

TRS/80
6 megabytes..... $399.95
10 megabytes $599.95
20 megabytes $799.95
30 megabytes $999.95
40 megabytes $1,299.95
5 megabyte removable w/30 or 40mb $399.95

Warranty Information: TURBO-M systems come with a full year warranty for parts and labor.
TURBO-M systems are guaranteed to run all programs such as LOTUS, SYMPHONY, AUTOCAD, FLIGHT SIMULATOR, DBASE, MICROGRAFX, WORDSTAR, LEADING EDGE W/P, RBASE.
MEGADISK hard drives are fully warranteed for parts and labor.

Software Support Inc. — MicroSmart Inc.

TERMS and CONDITIONS:
All prices are cash discounted. However, we do accept MC, VISA, AMEX & DISCOVER credit cards.
C.O.D.'s are accepted. No deposit required.
Purchase Orders-Corporate, Government & School P.O.'s are accepted. Please call for details.
Shipping Costs are calculated per order. Please call for total.
Shipments of all in-stock products are made within 24 hours. Same day service is available upon request — no added cost.
Not responsible for typographical errors.
Terms and Specifications may change without notice.

OUR NEW ADDRESS: 200 Homer Street Ashland MA 01721
OUR NEW COMPANY!
1-617-872-9090
Hours: Mon-Fri. 10 am to 5:30 pm (est) Sat 4:00 pm
Toll Free Ordering 1-800-343-8841
Dealer Inquiries Invited
Reader Service-Call 1-800-343-8841

Service & Returns. It is our policy to repair all service returns within 24-48 hours. Normally same day turn-a-round is accomplished. It is necessary to have a (R)eturn (M)aterial (A)uthorization to insure speed;y service.

IBM, TAVA, COLUMBIA, 5151, COMPAQ, EAGLE, TRS/80, HEATH/ZENITH are registered trademarks of IBM Corp., Tava Corp., Columbia Computer Corp., Key Tronics Corp., Compaq Corp., Eagle Computer, Tandy Corp., Zenith Corp., respectively
* 1986 Software Support, Inc. *All rights reserved*
* 1986 MicroSmart, Inc. *All rights reserved*

```
         *
         */

        if ( fgets(buffer,35,setup) == NULL )   {
            break;
        }
            cp = index(buffer,'\n');
            *cp = '\Ø';
            strcat(menuitems[i],buffer);
            if ( fgets(buffer,5Ø,setup) == NULL )   {
                scr_puts("Error reading menu command text:");
                            scr_printf("  Last line read was <%s>\n",menuitems[i]+3);
                exit(1);
            }
            cp = index(buffer,'\n');
            *cp = '\Ø';
            strcpy(menucommand[i],buffer);
        }

        strcat(menuitems[24],"Format Floppy Disk Drive");
        strcat(menuitems[25],"Execute a DOS Command");

        fclose(setup);

    }

    /*****************************************************************************
     *
     *      dead_screen()
     *          Performs two functions depending on the state of the variable
     *          "screenflag":
     *
     *          1.  If screen is on, it turns the screen off and initializes the
     *              walking prompt line.
     *
     *          2.  If the screen is off, it "walks" the flashing prompt to the
     *              next screen location with wraparound from bottom of the screen
     *              to the top.
     *
     *****************************************************************************/

    static int current_row;

    dead_screen()
    {
        if ( screenflag )   {
            screenflag = FALSE;
            current_row = -1;
            cls();
            if ( vflag == TANDY2ØØØ )   {
                normal = Øx2a;
                outportb(Øx198,Ø);
                outportb(Øx19c,Ø);
                outportb(Øx19e,Ø);
            }
            else    {
                normal = (normal & ØxØØØf) | ØxØØ88;
            }
        }
        else    {
            header(current_row,-1,8Ø," ",nor);
            if ( ++current_row > 23 )
                current_row = -1;
            header(current_row,-1,8Ø,"Press any key for menu",nor);
        }
    }

    /*****************************************************************************
     *
     *      drawmenu()
     *          Draw the menu box on the screen and display the proper menu text
     *
     *****************************************************************************/
```

```
#define UC  Ø
#define UR  Ø
#define LC  79
#define LR  24
#define LM  3

drawmenu()
{
    register int i;
    register int mrow;

    window(UR,UC,LR,LC,hnor);

    getdate();
    header(Ø,UC,LC,cmd,hrev);
    header(1,UC,LC,title,hrev);
    header(2,UC,LC,menuver,hrev);
    drawline(4,UC,LC,hnor);

    drawline(2Ø,UC,LC,hnor);
    header(2Ø,UC,LC," ",hrev);
    header(21,UC,LC,prompt,hrev);
    header(22,UC,LC," ",hrev);

    drawvert((LC-UC)/2,4,2Ø,hnor);

    attrib = hnor;
    for ( i = Ø, mrow=6; i < 13; i++, mrow++ )  {
        display(UC+LM,mrow,menuitems[i]);
        display(UC+(LC-UC)/2+LM,mrow,menuitems[i+13]);
    }
    screenflag = TRUE;
}

/*****************************************************************************
 *
 *      formatdisk()
 *          Called for option "Y" from the menu to display the option boxes
 *          on the video screen and get your response.  It then formats your
 *          response into a system call to format the floppy disk drive.
 *
 *****************************************************************************/

formatdisk()
{
    window(8,15,14,65,rev);
    display(17,9,"Format floppy disk in selected drive. Ensure");
    display(17,1Ø,"that a disk is in the drive with the door shut.");
    display(17,11,"Press the <F> key to format a data disk or");
    display(17,12,"press the <S> key to format a system disk. Any");
    display(17,13,"other key will return to the menu display.");
    ch = toupper(scr_getc());
    if ( (ch == 'F') || (ch == 'S') )   {
        window(1Ø,17,16,67,nor);
        display(19,12,"Press the 'A' or 'B' key to select drive: ");
        CURSOR_ON;
        do  {
            ch2 = toupper(scr_getc());
        }   while ( (ch2 != 'A') && (ch2 != 'B') );

        /*
         *  Note that this program assumes that FORMAT.COM has been renamed to
         *  FMT.COM on your disk.  Change the character string below to reflect
         *  the correct name if this is not your case.
         */

        if ( ch == 'F' )
            strcpy(cmd,"FMT A:/V");
        else
            strcpy(cmd,"FMT A:/S/V");
        cmd[4] = ch2;
        display(19,14,"Do you want to continue (Y/N)? ");
        do  {
```

```
            ch2 = toupper(scr_getc());
        }   while ( (ch2 != 'Y') && (ch2 != 'N') );
        if ( ch2 == 'Y' )
            doscommand(cmd);
        else
            CURSOR_OFF;
    }
    screenflag = FALSE;
}

/*****************************************************************************
 *
 *      doscommand(str)
 *          Restores the video to normal and executes the user command
 *          specified by parameter "str"
 *
 *****************************************************************************/

doscommand(str)
char *str;
{
    cls();
    scr_call(SMODE);
    CURSOR_ON;
    system(str);
    if ( vflag == TANDY2000 )
        init_video();
    screenflag = FALSE;
}

/*****************************************************************************
 *
 *      get_dos_command()
 *          This function is executed in response to option "Z" from the
 *          system menu.  It draws the "DOS command box" on the screen and
 *          reads the user command input.  If the input is non-null, the
 *          command is executed.  When the command completes, the screen is
 *          intact, waiting for you to press a key to return to the menu.
 *
 *****************************************************************************/

get_dos_command()
{
    register char *cp;
    register char c;
    int row,col;

    window(12,0,17,79,nor);
    header(12,0,79,"DOS COMMAND WINDOW",rev);
    header(13,0,79,"Enter DOS command below, ENTER to return to menu",rev);
    CURSOR_ON;
    cp = cmd;
    *cp++ = '$';
    for (;;)    {
        *cp = 0;
        display(1,16,cmd);
        c = scr_getc();
        if ( c == '\r' )
            break;
        if ( c == 8 )   {
            if ( cp > cmd+1 )   {
                cp--;
                scr_loc(&row,&col);
                display((--col),row," ");
            }
            continue;
        }
        if ( cp > cmd+79 )
            continue;
        if ( (c & 0xff) < 0x20 )
            continue;
        *cp++ = c;
    }
    if ( cmd[1] == 0 )  {
```

```
            CURSOR_OFF;
            screenflag = FALSE;
            return(Ø);
        }
        cls();
        scr_call(SMODE);
        system(cmd+1);
        header(23,-1,8Ø,"Press any key to return to the menu",rev);
        scr_getc();
        if ( vflag == TANDY2ØØØ )
            init_video();
        screenflag = FALSE;
    }

    /*****************************************************************************
     *
     *      getdate()
     *          This function reads the current system date using MS-DOS function
     *          call 2AH and formats the returned date in the string buffer array
     *          "cmd" for display.
     *
     *****************************************************************************/

    char *weekdays[] = {
        "Sunday",
        "Monday",
        "Tuesday",
        "Wednesday",
        "Thursday",
        "Friday",
        "Saturday"
    };

    char *months[] = {
        "",
        "January",
        "February",
        "March",
        "April",
        "May",
        "June",
        "July",
        "August",
        "September",
        "October",
        "November",
        "December"
    };

    getdate()
    {
        struct regs dosregs;

        dosregs.AX = Øx2aØØ;
        sysint(33,&dosregs,&dosregs);
        sprintf(cmd,"%s, %d %s %4d",weekdays[dosregs.AX & Øxff],
            (dosregs.DX & Øxff),months[((dosregs.DX >> 8) & Øxff)],dosregs.CX);
    }

    /*****************************************************************************
     *
     *      gettime()
     *          This function gets the current system time using MS-DOS function
     *          call 2CH and returns a long integer value representing the
     *          binary number of seconds past midnight.  As a side-effect, this
     *          function also formats the time in display format in the character
     *          string buffer "cmd".
     *
     *****************************************************************************/

    long gettime()
    {
        struct regs dosregs;
        int hours,minutes,seconds;
```

```
        dosregs.AX = 0x2c00;
        sysint(33,&dosregs,&dosregs);
        hours = (dosregs.CX >> 8) & 0xff;
        minutes = dosregs.CX & 0xff;
        seconds = (dosregs.DX >> 8) & 0xff;
        sprintf(cmd,"%2d:%2d:%2d",hours,minutes,seconds);
        if ( cmd[3] == ' ' )
            cmd[3] = '0';
        if ( cmd[6] == ' ' )
        cmd[6] = '0';
    return( (long)seconds + 60L * ( (long)minutes + 60L * (long)hours) );
}

/*****************************************************************************
 *
 *  Init_video - initialize the video segment address to the correct
 *  value.
 *
 *  The IBM-PC and clone's video segment is dependent on the hardware
 *  installed.  Monochrome VRAM begins at segment 0xB000 and color VRAM
 *  begins at segment 0xB800.
 *
 *  The TANDY 2000 video segment is located in the highest 5K of memory
 *  installed.  To determine if the computer is a Tandy 2000, you must fir

st
 *  inspect the Boot ROM located beginning at 0xFC00 segment for the prope

r
 *  identification.  There are currently two versions of this Boot ROM with
 *  different locations of the Tandy identifier.  This explains the relatively
 *  complex "if" statement below.
 *
 *  After determining the segment addresses correctly, the colors and attri-
 *  butes must be set correctly.  On the IBM-PC and clones, this is the same --
 *  colors and attributes are controlled by the same byte.  Tandy 2000 mon

o-
 *  chrome attributes are totally different from the IBM-PC.
 *
 *****************************************************************************/

init_video()
{
    struct regs memregs;

    if ( ( (peekb(0x002f,0xfc00) == 'T') && (peekb(0x0030,0

/xfc00) == 'a') )
      || ( (peekb(0x0032,0xfc00) == 'T') && (peekb(0x0033,0x

fc00) == 'a') ) )
    {
        vflag   = TANDY2000;
        on_value  = 0x000F;
        off_value = 0x6000;
        normal  = 0x0a;
        revrs   = 0x8a;
        hilite  = 0x4a;
        rhilite = 0xca;
        SMODE = 2;
        scr_call(SMODE);
        outportb(0x198,NORMAL);
        outportb(0x19a,REVRS);
        outportb(0x19c,HILITE);
        outportb(0x19e,RHILITE);
        sysint(18,&memregs,&memregs);   /* get current memory size */
        Video_Mem = ( (memregs.AX << 6) & 0xFF00 ) | 0x00C0;
    }
else    {
    on_value  = 0x0007;
    off_value = 0x3000;
    sysint(17,&memregs,&memregs);   /* get equipment status word */
```

Give your Model 100
128K RAM

Software included, transfers from bank to bank.

~~$425~~
sale ends July 31 1986
$349.00

Works like main menu! Includes powerful RAM Basic that lets programs store and access data from any other bank.

As amazing as it seems you can upgrade your Model 100 to 128K of RAM in just 60 seconds.

It comes right out of the box looking just like the picture. You just open the little compartment on the back of your Model 100 with a quarter and it just pushes right into place. You can then put the cover back in its place.

You then have 4 banks of RAM of 32K each. The additional three banks also work just like your Main Menu.

You push a function key and you are in the second bank. Push again and you are in third, again, then fourth. Press it once again for your original bank.

It has its own built-in NiCad battery that recharges right from the Model 100 and its guaranteed for a full year.

What is really great is that you can copy a file from one bank to another with just a function key.

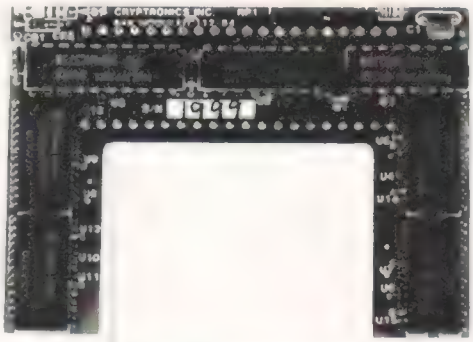
Each bank is like having another Model 100, and all the built-in programs as well as any snap-in ROM programs appear in all four banks and work the same way. Your widebar cursor moves from file to file and you access any file or run any program just by pressing ENTER.

What lets you copy any file from one bank to another is a snap-in ROM from PCSG called RAM+, that comes at no extra charge. It pushes right into the little socket in that same compartment with the 96K expansion unit.

Not only does this firmware let you copy a file from bank to bank, but you can make a copy of any file within the same bank instantly with a function key. Great for Lucid spreadsheets!

Copy a file from bank to bank with a function key

You can also rename a file, or kill any file with just a function key. Plus you can do a whole lot of other useful things like setting the date, day and time with function key ease. You even have a function key that lets you use non-Radio Shack printers without having to make those tricky dipswitch settings.



Installs as easily as plugging in a socket

RAM+ lets you cold start any one of your banks without affecting the other three. That means that anytime you want you can clean out a bank's entire memory, but leave intact all the files in the other banks.

What is also fantastic is that you don't have to have the ROM in place to use the additional RAM. Whenever you take out the snap-in ROM it leaves behind a tiny machine code program that lets you switch from bank to bank just by pressing ENTER.

This lets you use your ROM socket to snap-in other ROMS like the amazing SUPER ROM, or DISK + ROM file transfer program, and use them in any or all four banks. All of these, by the way, are available from PCSG.

When you are ready to copy a file from one bank to another or use any of the other fantastic functions we talked about you can just snap the RAM + ROM back into place.

Everybody that has this 128K system in their Model 100 is so excited, because it gives them four times the capacity and all banks work just like the Main Menu.

And what has made a lot of people happy is that the system bus, located in the same compartment, is left free for you to plug in a DVI or the Holmes Engineering/PCSG portable disk drive.

The ability to copy a file from bank to bank instantly with a function key, plus all of the other features make this RAM extension truly an engineering masterpiece.

This is the only memory expansion that gives you 3 extra banks of RAM. That's 50% more than any other RAM expansion product. Thousands have been shipped to satisfied customers over the past 12 months — more are installed and working reliably than any competitive product.

Some people hesitate when they think of installing something, and then others are skeptical that any additional hardware could be as good as the Model 100 itself. That's why we sell these 96K expansions on a 30 day trial. Simply return it within 30 days for a full refund if you are not satisfied. Priced at ~~$425~~. MC VISA COD.

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd. #207
Dallas, Texas 75229
214-351-0564

PCSG offers a 30-day/money-back trial! • Also available for the model 200.

```
        if ( (memregs.AX & 0x0030) == 0x0030 )  {
            vflag = IBMMONO;
            Video_Mem = 0xB000;
            SMODE = 2;
            scr_call(SMODE);
            normal  = 0x07;
            revrs   = 0x70;
            hilite  = 0x0f;
            rhilite = 0x7f;
        }
        else    {
            vflag = IBMCOLOR;
            Video_Mem = 0xB800;
            scr_call(SMODE);
            normal  = NORMAL;
            revrs   = REVRS;
            hilite  = HILITE;
            rhilite = RHILITE;
        }
    }
    attrib = normal;
    cls();
    CURSOR_OFF;
}

/*****************************************************************************
 *
 *  Draw a "windowed box" on the video screen in the area specified
 *  by the upper left corner and the lower right corner.  The box must
 *  be at least 3 X 3 to accomodate the frame drawn around it.
 *
 *  The window box will be drawn using a double-line frame in the attribute
 *  mode specified.  The character attibute will be restored to its previous
 *  value on exit from the procedure.
 *
 *****************************************************************************/

#define ULC     0xc9                /* Define box characters using the  */
#define URC     0xbb                /* IBM-PC double line special chars */
#define LLC     0xc8                /* You can change these to the      */
#define LRC     0xbc                /* smaller single line characters   */
#define HRZ     0xcd                /* or any other character you       */
#define VRT     0xba                /* desire.                          */
#define LVR     0xcc
#define RVR     0xb9
#define UHZ     0xcb
#define LHZ     0xca

static char window_line[81];

window(urow,ucol,lrow,lcol,atr)
int ucol, urow, lcol, lrow, atr;
{
    register int line_len, i, j;

    if ( ((line_len = lcol-ucol+1) < 3) || (line_len > 80) )
        return(-1);
    if ( (lrow-urow+1 < 3) || (lrow-urow+1 > 25) )
        return(-1);
    if ( (atr < 1) || (atr > 4) )
        return(-1);

    attrib = atr;

    window_line[0] = ULC;
    window_line[line_len-1] = URC;
    for ( i = 1; i < line_len-1; i++ )
        window_line[i] = HRZ;
    window_line[line_len] = '\0';
    display(ucol,urow,window_line);

    window_line[0] = window_line[line_len-1] = VRT;
    for ( i = 1; i < line_len-1; i++ )
        window_line[i] = ' ';
```

```
    window_line[line_len] = '\Ø';
    for( i = 1; i < lrow-urow; i++ )    {
        display(ucol,urow+i,window_line);
    }

    window_line[Ø] = LLC;
    window_line[line_len-1] = LRC;
    for ( i = 1; i < line_len-1; i++ )
        window_line[i] = HRZ;
    window_line[line_len] = '\Ø';
    display(ucol,lrow,window_line);

    return(Ø);
}

/*****************************************************************************
 *
 *  header(row,col,ecol,str,at)
 *          displays the string "str" as a header centered one line down
 *          from the top row of the window defined by the top row and the
 *          starting and ending columns.  This actually can be used to center
 *          a line anywhere on the screen and write it using the attributes
 *          desired.
 *
 *****************************************************************************/

header(row,col,ecol,str,at)
int col, row, ecol, at;
char *str;
{
    char buf[81];
    register int i, len;
    int atsave;

    atsave = attrib;
    attrib = at;
    len = ecol - col - 1;
    for ( i = Ø; i < len; i++ )
        buf[i] = ' ';
    for ( i = (len - strlen(str))/2; *str; )
        buf[i++] = *str++;
    buf[len] = '\Ø';
    display((++col),(++row),buf);
    attrib = atsave;
}

/*****************************************************************************
 *
 *      drawline()
 *          Draw a horizontal line across the screen with proper vertical
 *          line terminators.
 *
 *****************************************************************************/

drawline(row,ucol,lcol,atr)
int row,ucol,lcol,atr;
{
    char linestr[81];
    register int i;
    int attsave;

    linestr[Ø]           = LVR;
    linestr[lcol-ucol]   = RVR;
    linestr[lcol-ucol+1] = Ø;

    for ( i = 1; i < (lcol-ucol); i++ ) {
        linestr[i] = HRZ;
    }

    attsave = attrib;
    attrib = atr;
    display(ucol,row,linestr);
    attrib = attsave;
}
```

```
/*****************************************************************************
 *
 *      drawvert()
 *          Draw a vertical line down the screen with proper horizontal line
 *          terminators.
 *
 *****************************************************************************/
drawvert(col,urow,lrow,atr)
int col,urow,lrow,atr;
{
    register int i;
    int attsave;

    attsave = attrib;
    attrib = atr;
    cmd[Ø] = UHZ;
    cmd[1] = Ø;
    display(col,urow,cmd);
    cmd[Ø] = LHZ;
    cmd[1] = Ø;
    display(col,lrow,cmd);
    cmd[Ø] = VRT;
    cmd[1] = Ø;
    for ( i = ++urow; i < lrow; i++ )
        display(col,i,cmd);
    attrib = attsave;
 }
/*****************************************************************************
 *
 *      cls()
 *          Clear the video screen and home the cursor to the upper left
 *          screen corner
 *
 *****************************************************************************/
cls()
{
    CLS;
    scr_curs(Ø,Ø);
}
/*****************************************************************************
 *
 *  display(x,y,str)    --- displays the string pointed to by the variable
 *                      "str" beginning at the cursor location specified by
 *                      (x,y).  Uses direct video memory access.  The chars
 *                      are displayed in the colors determined by the value
 *                      of "attrib".
 *
 *****************************************************************************/
display(x,y,str)
int x, y;
char *str;
{
    register int len, atr;
    switch (attrib) {
        case 1:
            atr = normal;
            break;
        case 2:
            atr = revrs;
            break;
        case 3:
            atr = hilite;
            break;
        case 4:
            atr = rhilite;
            break;
        default:
            atr = normal;
    }
    len = strlen(str);
    for ( ; len-- ; )   {
/*
 *  The next segment of code is conditionally assembled here for an IBM-PC to
```

```
        *  ensure that the characters are only written to the video monitor during the
        *  video retrace mode or flyback.  This code should work on all close
        *  compatibles.
        *
        */
              if ( vflag == IBMCOLOR )    {
                  #asm
                       MOV     AX,4ØH      |       ;Set communications segment address
                       PUSH    DS          |       ;  ..  in DS register
                       MOV     DS,AX
                       MOV     DX,[63H]    |       ;Get 6845 Video Controller Base addr
                       ADD     DX,6                ;Form status port address
                       POP     DS
                  __L1: IN     AL,DX               ;Get controller status
                       TEST    AL,1                ;Is horizontal retrace low?
                       JNZ     __L1                ;Wait until it is
                       CLI                         ;Disable interrupts until write compl
                  __L2: IN     AL,DX               ;Test for retrace high
                       TEST    AL,1
                       JZ      __L2                ;Wait until it is
                       MOV     AX,5                ;Small delay loop
                  __L3: DEC    AX                  ; .. so hash is not written at left
                       JNZ     __L3                ; .. margin of screen
                    #endasm
              }
              pokew((y * 8Ø + x++) << 1,Video_Mem,(atr << 8) | (*str++ & ØxØØf

      f));
              if ( vflag == IBMCOLOR )    {
                  #asm
                      STI                 |       ;Reenable interrupts
                    #endasm
              }
          ]
          scr_curs(y,x);
      }
```

PCM

SPECIAL

TANDY 1000

Multifunction Board

- 512 K Ram
- RS-232C Serial Port
- DMA Controller
- Clock-Calendar

$259.95

ADDITIONAL TANDY COMPATIBLE
PRODUCTS AVAILABLE
CALL FOR A FREE CATALOG

Micro Mainframe 4N1®

11285-E Sunrise Gold Circle
Rancho Cordova, CA 95670

(916) 635-3997

VISA

LOW LOW PRICES
on
Hard Disk Drives
Floppy Disk Drives
Star Printers
Anchor Modems
Diskettes
Color Computer Drø
SPECIAL

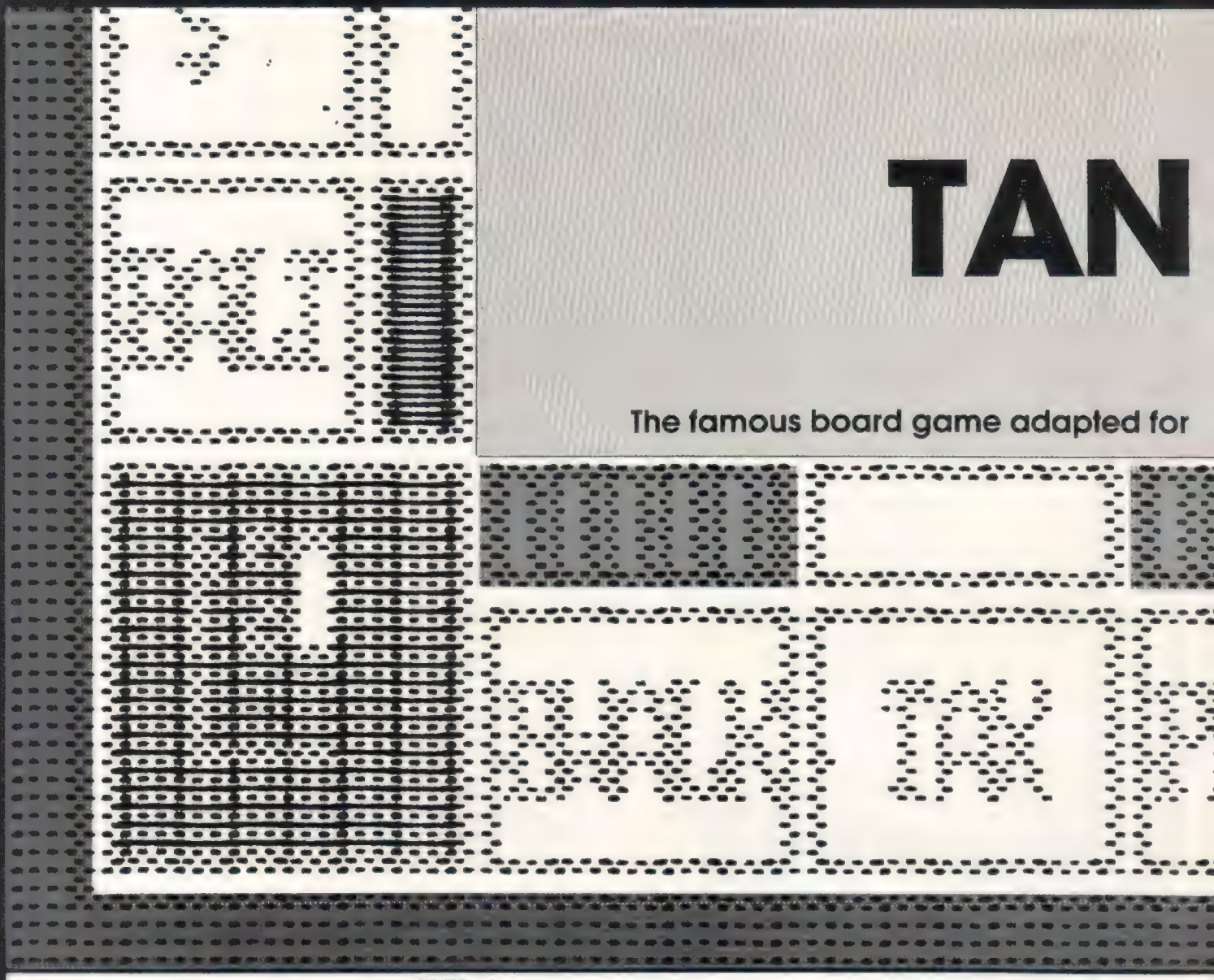
IBM XT COMPATIBLE
2 FLOPPY DRIVES
1 10 MEG HARD DISK
MONOCHROME OR COLOR CARD
256 K MEMORY EXPANDIBLE TO 640 K
$1195.00

CALL US TODAY!! 617-278-6555 • DEALER INQUIRIES INVITED.
ORDER TOLL FREE 1-800-635-0300 617-278-6555

TRUE DATA PRODUCTS
9 South Main St.
Uxbridge, Mass 01569
(617) 234-7047
We welcome Visa/Master Charge
Checks (allow 2 weeks for clearing)
• COD Add $2.00
HOURS MON-SAT 9-8 (EST)
617-278-6555

TAN

The famous board game adapted for



Tanyopoly is a computerized take-off on an old traditional board game. Your Tandy 1000 BASIC is a powerful implementation of Microsoft BASIC. It provides all the tools needed to create a game which can provide hours of pleasure for the entire family. The full playing board is used in this version of the game, complemented by graphically rolling dice, moving playing pieces, sound effects, a window for quick summaries and a banker who never makes a mistake with your money!

Tanyopoly requires a minimum of 256K of memory in your Tandy 1000. The program is long, but it didn't start out to be such a big project. Friends and family kept adding ideas for special features, and pretty soon I was looking at over 30K of BASIC. Once added, I couldn't bear removing any of the goodies. It should be possible to modify *Tanyopoly* to run on the 1200, 2000 and 3000 machines. To do so, change the SCREEN 6 color display to a SCREEN 2 monochrome version. Remove the CLEAR 32768 statement and PALETTE statements. Other changes would be of a minor nature.

Here is a suggestion as to the order in which the listing should be typed in. First type in lines 1-320, then skip to 2985 and do all the lines following. If you then insert a temporary line at 340 (340 GOTO 340), you can test and debug the playing screen completely, before going into the playing portion of the program. When all is working, remove the *temporary* Line 340 and type in the rest of the program. If you are a novice, you should be aware that the balance of the program will not run until virtually all of it is completed.

The first section of the program is the title screen, set up to provide a bit of color and flash to introduce the game. I have chosen to create my own letters for the title, as seen in lines 130 and 135. These are put in place on the screen using the DRAW command to overlap three sets of letters in different colors. This gives a nice look to the title, which is then surrounded by a flashing marquee created using the STYLE option of LINE (Line 180).

The main program is set up in the area from lines 195-424. We must carefully dimension all the array space that will be needed to get dice, windows, playing pieces, etc. I have liberally sprinkled the listing with REM statements so you can follow most of the subroutines which make up the program. I will not be going into great

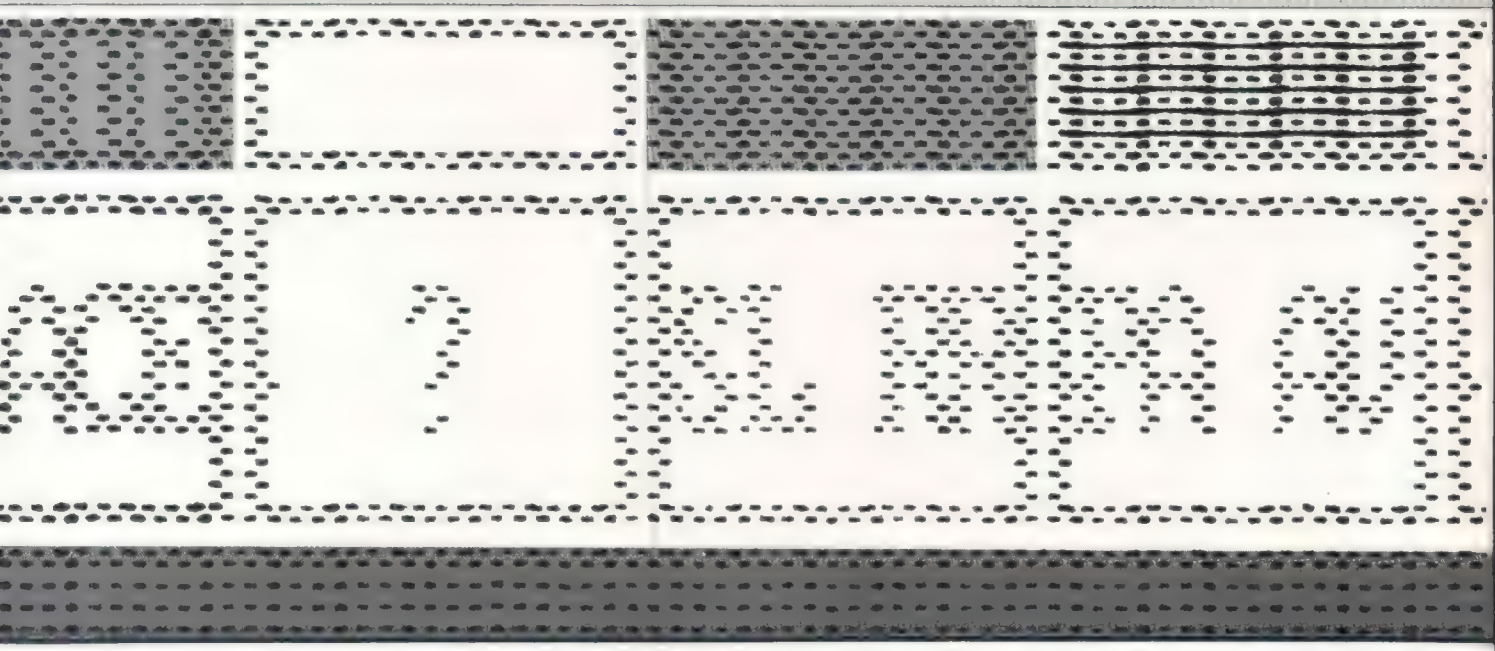
---

Leonard Hyre works as a claims representative for the Social Security Administration. He has written several articles for THE RAINBOW, PCM's sister publication for the Color Computer, and is the author of a number of commercial programs. He may be contacted at P.O. Box 403, Cambridge, MD 21613; (301) 228-0064.

YOPOLY

your Tandy 1000

By Leonard Hyre



detail about some of the program structure, however, some of the subroutines are unique and may be of interest to the programmers out there.

The core of the program, in lines 405 through 550, keep track of who's turn it is and where players are on the board, sends the player off to special features ("Free Parking," "Income Tax," etc.), and changes players at the end of a roll unless a double is rolled. All other routines lead back to the core.

Rolling the graphic dice is the next subroutine. The dice, having been created way down in lines 3315-3335, are manipulated in lines 560 through 645, putting the dice obtained with the earlier GET, in the proper location on the screen. When a player presses any key, the roll of the dice ceases, a final set is put on screen and the player is told what he has rolled!

Property! Property! Who owns it? Can I buy it? Do I have a hotel on it? Control of money and property is the *thing*. Several subroutines are used to control these various functions individually. There are subroutines for "Buy" (785-900), "Book Status" (850-900), "Improve Property" (910-1110), "Asset Listing to Screen or Printer" (1120-1235), "Mortgage" (1245-1360), "Unmortgage" (1370-1445) and "Rent Due" (1470-1525). All of these are similar because they use the information read into memory earlier, to control just who can do what and when.

For example, the Buy routine first checks (in lines 800-805) to see if the players location is one that can't be bought or sold (such as "Jail") by checking to see if O(*X*) is 5 (*X* = Location and 5 indicates "not for sale." If this is true, "You can't buy that!" is shown to the potential buyer, and the routine ends. If not true, the routine continues. If the bank is the owner (O(*X*)=0), the property is sold to the purchaser. Otherwise, the property belongs to the other guy or already belongs to the player. If either is true, an announcement is made (lines 820-835) and the routine finishes with GOTO 720. At that time GOSUB 860 is made to update "Book Status" and the next option is offered to the player. Other subroutines employ the same type of logic to keep track of their particular jobs. Follow through the lisitng if the "hows and whys" hold your interest.

Three particular routines use the GET and PUT commands, and generally give novice programmers fits! We use GET to save a portion of the graphic screen in memory until we wish to put it on the screen where we choose. For example, in the Summary Window routine (lines 2850 through 2965), we get a rectangular area first (Line 2960) and save it as V. We then use a simple LINE statement to black out that same area as window effect and print any information we choose on it. When done, we just have to put back the area we got with the GET. See Line 2960 for the command. The

PSET Option tells BASIC to put the array back exactly as it was. The dice routine uses the same technique as does the moving playing pieces.

Also of possible interest are the sound effects. A siren accompanies the inmate in the "go to jail" scenarios. Landing on an opposing player's railroad will illicit an approaching train horn. Arriving in free parking, you get tooted at before being awarded the "pot." The Jail routine is at lines 2200-2280, the Railroad Fees at 2365-2430 and the Free Parking from 2290 through 2310.

Two special notes concerning typing in the program must be made. Line 210 has a statement, CLEAR 32768. *Do not type in the exclamation mark* as seen in the listing. It will magically appear on its own. The other tricky typing proposition is the playing board borders (lines 3255-3290). They are created on the Tandy 1000 by holding down the ALT key and typing the ASCII code for the particular character needed. When you release the ALT key the character will appear. I have given you the codes for the needed characters in the remarks just above this portion of the listing. Be sure to get the proper number of spaces in the lines here or your board will be a scrambled egg or worse!

Ah! At last it's finished. *Tanyopoly* has some special rules which you should be aware of before playing. I will summarize some of them here.

There is no limit to the number of "Doubles" you can roll on your turn.

When you land on a property and rent is due your opponent, the program automatically takes the money away from you and gives it to the property owner. If this rent due is greater than your assets and cash, you are bankrupt and the winner is announced. If your cash and assets are great enough to cover your debt, but you are short of cash, the program will force you to continue to mortgage property until you have a positive amount of cash.

A fast version of the game is offered before play begins. If you choose, the program will randomly deal out the property to the two players or teams. This leads to a much faster game. However, if one player gets two or three books and the other player gets zilch, please don't blame me. It's just the hand of fate. You can always reason it out with each other (sure you can).

In jail, you do not lose a turn. If you don't have a Get-Out-Free credit, you must cough up the $50.

When buying houses or hotels, each piece of property in the book must be improved equally.

If the typing is overwhelming, I'll be glad to provide you with the program on a disk for $6.00 to cover costs (L. Hyre, P.O. Box 403, Cambridge, MD 21613). Alternatively, by the time you read this, PCM On Disk should be available, giving you all the MS-DOS programs in the month's issue. If you encounter problems, feel free to contact me in writing, by phone or by leaving a message to MUNCH in the Delphi MSDOS Special Interest Group, sponsored by PCM.

By all means, have fun and be careful or you just might "go to jail!" □

The listing:

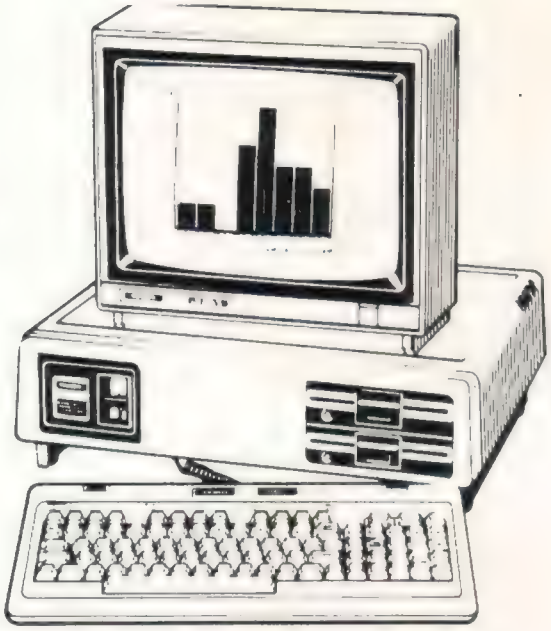
```
1 '*********************************************************************
2 '*   TANYOPOLY...A Board Game for the TANDY  1ØØØ  Computer          *
3 '*                    (C) 12/85 , by L.Hyre                          *
4 '*                                                                   *
5 '*                 > 256K or More REQUIRED! <                        *
6 '*                                                                   *
7 '*********************************************************************
8 '
9 '
1ØØ '*** TITLE SCREEN ***
1Ø5 '
11Ø KEY OFF:SCREEN 1:CLS
115 COLOR Ø:PALETTE 1,14:PALETTE 2,12:PALETTE 3,9
12Ø '
125 '*** Create Special Lettering ***
126 '
13Ø XT$="U2ØL6U4R18D4L6D2ØL6":XA$="U24R16D24L4U1ØL8D1ØL4":XN$="U24R4F8U8R4D24L4H
8D8L4"
135 XY$="U8H8U8R4D8R16U8R4D8G8D8L8":XO$="U24R16D24L16":XP$="U24R16D12L12D12L4":X
L$="U24R4D2ØR12D4L16"
14Ø XTAN$="xxt$;BR18;XXA$;BR25;XXN$;BR32;XXY$;BR24;XXO$;BR24;XXP$;BR24;XXO$;BR24
;XXL$;BR24;XXY$;"
145 DRAW"BM54,1ØØ;xxtan$;":DRAW"BM56,1Ø1;C2;XXTAN$;":DRAW"BM58,1Ø2;C1;XXTAN$;"
15Ø LINE(2Ø,6Ø)-(3ØØ,13Ø),16,B
155 LOCATE 15,16:PRINT"by L. Hyre":CX=1
16Ø '
165 '*** Flashing Marquee ***
17Ø '
175 FOR FLASH=1 TO 3:B1=18:B2=58:B3=3Ø2:B4=132
18Ø FOR BOX=1 TO 8:LINE(B1,B2)-(B3,B4),CX,B,&H5555:B1=B1-2:B2=B2-2:B3=B3+2:B4=B4
```

```
+2:CX=CX+1:IF CX=4 THEN CX=CX-3
185 NEXT BOX,FLASH
19Ø '
195 '*** MAIN PROGRAM SETUP HERE ***
2ØØ '
2Ø5 '*** Note: DO NOT Type in the EXCLAMATION MARK in next line !!!!! ***
21Ø CLEAR ,,,32768!
215 '
22Ø '*** Reserve Space for Arrays ETC ***
225 '
23Ø DIM V(234Ø),U(33),W(33),NU(76),NV(76),NW(76),NX(76),NY(76),NZ(76)
235 DIM P$(4Ø),O(4Ø),RØ(4Ø),R1(4Ø),R2(4Ø),R3(4Ø),R4(4Ø),R5(4Ø),HC(4Ø),MV(4Ø),UN(
4Ø),C1$(12),C2$(12),C4$(12),C5$(12),RL1(12),RL2(12),P(4Ø),H(4Ø),PROP(4Ø),B(4Ø),U
PROP(4Ø),O1(4Ø),D1(4Ø)
24Ø '
245 RANDOMIZE TIMER
25Ø L1=Ø:L2=Ø:TH(1)=Ø:TH(2)=Ø:JF(1)=Ø:JF(2)=Ø:JB(1)=Ø:JB(2)=Ø:RROWN(1)=Ø:RROWN(2
)=Ø:FM=1ØØ:PRP(1)=Ø:PRP(2)=Ø:DB=Ø:M(1)=15ØØ:M(2)=15ØØ
255 FOR XX=1 TO 4Ø:H(XX)=Ø:NEXT
26Ø '
265 '
27Ø '*** Set Up Game For Play
275 '
28Ø GOSUB 2985:'*** Read in Property ***
285 GOSUB 3485:'*** Read in Chancey and Chest ***
29Ø GOSUB 354Ø:'*** Read in Playing Piece Positions
295 '
3ØØ KEY OFF:SCREEN 6:COLOR 1,Ø
3Ø5 PALETTE 1,15:PALETTE 2,4:PALETTE 3,9
31Ø CLS
315 GOSUB 3255
32Ø '
325 '
33Ø '
335 '
34Ø '** SET UP OPTIONS AND INITIAL VARIABLES **
345 '
35Ø DN=6:FOR N=1 TO 2:LOCATE DN,17,1:INPUT"PLAYER NAME:";N$(N):DN=DN+1:NEXT N
355 MID$(N$(1),1,1)=CHR$((ASC(LEFT$(N$(1),1)) AND 95))
36Ø MID$(N$(2),1,1)=CHR$((ASC(LEFT$(N$(2),1)) AND 95))
365 IF LEN(N$(1))>7 THEN N$(1)=LEFT$(N$(1),7)
37Ø IF LEN(N$(2))>7 THEN N$(2)=LEFT$(N$(2),7)
375 LOCATE 11,17:INPUT"Fees Paid Into FREE PARKING (Y/N)";FP$:IF FP$="Y" OR FP$=
"y"THEN FP=1 ELSE IF FP$="N"OR FP$="n"THEN FP=2 ELSE BEEP:LOCATE 11,51:PRINT"
":GOTO 375
38Ø LOCATE 12,17:INPUT"Pass Out The Properties ?   (Y/N)";PO$:IF PO$="y"OR PO$="
Y" THEN 2565
385 GOSUB 69Ø:LOCATE 11,17:PRINT"Let's play TANYOPOLY!";:LOCATE 12,17:PRINT"OK..
.I'll roll to see who goes first!":FOR DL=1 TO 1ØØØ:NEXT DL:FOR PICK=1 TO 15:SOU
ND INT(RND(1)*5ØØ),.8:FOR DL=1 TO 15:NEXT:NEXT PICK
39Ø RL1=INT(RND*6)+1:RL2=INT(RND*6)+1:IF RL1=RL2 THEN 39Ø
395 IF RL1>RL2 THEN TR=1 ELSE TR=2
4ØØ LOCATE 13,17:PRINT N$(TR);" goes first!";
4Ø5 FOR DL=1 TO 15ØØ:NEXT:GOSUB 665:GOSUB 69Ø
41Ø LOCATE 6,17:PRINT"PLAYER:";N$(1):LOCATE 6,47:PRINT"PLAYER:";N$(2);
415 LOCATE 7,17:PRINT"CASH $";M(1):LOCATE 7,47:PRINT "CASH $";M(2)
42Ø LOCATE 3,79:COLOR 3:PRINT PP1$;:LOCATE 9,79:COLOR 2:PRINT PP2$;:COLOR 1
425 GET(Ø,Ø)-(15,8),U:GET(Ø,Ø)-(15,8),W
43Ø '
```

NOCONA

ELECTRONICS

TRS-80 Model 2000 by Tandy®



MS-DOS Based

Computer System

256K MOD2000 W/MONO MONITOR	898.00
640K MOD3000 W/35 MEG HD	3549.00
256K MOD 1200 HD 2 DR.	1499.00
128K MOD 1000 W/MONO 1 DR.	719.00
128K MOD 1000 W/COLOR 2 DR.	1029.00
24K MOD 102 PORT	369.00
24K MOD 100 PORT	319.00
24K MOD 200 PORT	679.00
64K MOD 4D 2 DR	839.00
64K COLOR COMPUTER	139.00
512K MOD 6000 W/15MEG HD	2579.00
70 MEG HARD DISK	2679.00
35 MEG HARD DISK	2059.00
DMP 105 PRINTER	145.00
DMP 2200 PRINTER	1149.00
DMP 130 PRINTER	249.00
DMP 430 PRINTER	499.00
DMP 2100P PRINTER	749.00
CGP 220 COLOR PRINTER	449.00
DWP 510 PRINTER	549.00
TRACTOR DWP 220	99.00
TRACTOR DWP II/510	159.00
TRACTOR FEED DMP2100P	99.00
512K MOD 3000 W/20 MEG HD	2499.00
PARALLEL PRINTER CONTROLLER	179.00
PARALLEL PRINTER CABLE	29.00
DISK DRIVE, MOD 100-200	160.00
32K MOD 600 PORTABLE	1149.00
512K MOD 3000 1 DR.	1819.00

TRS-80® Model 100 by Radio Shack

100% RS COMPONENTS NO FOREIGN DRIVES OR MEMORY FULL WARRANTY
ALL RS SOFTWARE 20% OFF CATALOG PRICE
CASHIERS CHECK OR MONEY ORDER MUST ACCOMPANY ALL ORDERS

(817) 825-4027

NOCONA ELECTRONICS • BOX 593 • NOCONA, TX 76255

```
435 '***** WHO'S TURN IS IT ANYWAY AND WHERE IS HE ****
44Ø '
445 LOCATE 8,3Ø:PRINT STRING$(22," ");:GOSUB 69Ø
45Ø LOCATE 8,3Ø:PRINT N$(TR);"'S  Turn."
455 LOCATE 11,17:IF TR=1 THEN PRINT "YOU'RE AT ";P$(L1) ELSE PRINT "YOU'RE AT ";
P$(L2)
46Ø IF TR=1 AND L1=Ø THEN LOCATE 11,17:PRINT"YOU'RE AT GO!";:ELSE IF TR=2 AND L2
=Ø THEN LOCATE 11,17:PRINT"YOU'RE AT GO!";
465 GOSUB 57Ø:LOCATE 13,17:IF TR=1 THEN PRINT"YOU MOVE TO ";P$(L1);ELSE IF TR=2
THEN PRINT"YOU MOVE TO ";P$(L2)
47Ø IF FP=1 AND (TR=1 AND L1=2Ø) OR (TR=2 AND L2=2Ø) THEN 23ØØ
475 IF (TR=1 AND L1=3Ø) OR (TR=2 AND L2=3Ø) THEN 221Ø
48Ø IF (TR=1 AND L1=4) OR (TR=2 AND L2=4) THEN GOSUB 2115
485 IF (TR=1 AND L1=38) OR (TR=2 AND L2=38) THEN GOSUB 2165
49Ø IF (TR=1 AND (L1=2 OR L1=17 OR L1=33)) OR (TR=2 AND (L2=2 OR L2=17 OR L2=33)
) THEN GOTO 1545
495 IF (TR=1 AND (L1=7 OR L1=22 OR L1=36)) OR (TR=2 AND (L2=7 OR L2=22 OR L2=36)
) THEN GOTO 1685
5ØØ IF TR=1 AND O(L1)=2 AND (L1=5 OR L1=15 OR L1=25 OR L1=35) THEN GOSUB 2375:GO
TO 525
5Ø5 IF TR=2 AND O(L2)=1 AND (L2=5 OR L2=15 OR L2=25 OR L2=35) THEN GOSUB 2375:GO
TO 525
51Ø IF TR=1 AND O(L1)=2 AND (L1=12 OR L1=28) THEN GOSUB 2455:GOTO 525
515 IF TR=2 AND O(L2)=1 AND (L2=12 OR L2=28) THEN GOSUB 2455:GOTO 525
52Ø IF (TR=1 AND O(L1)=2) OR (TR=2 AND O(L2)=1) THEN GOSUB 147Ø
525 LOCATE 13,45:COLOR 2:PRINT"<Any Key>";:COLOR 1
53Ø AK$=INKEY$:IF AK$="" THEN 53Ø
535 GOTO 72Ø
54Ø IF TR=1 THEN TR=TR+1 ELSE IF TR=2 THEN TR=TR-1:'*** CHANGE PLAYERS HERE!
545 IF DB=1 THEN GOSUB 283Ø:'*** HEY! THAT'S A DOUBLE
55Ø GOTO 445
555 '
56Ø '*** ROLL DICE ROUTINE **
565 '
57Ø RL1=INT(RND(1)*6)+1:RL2=INT(RND(1)*6)+1
575 IF TR=1 AND L1+(RL1+RL2)=L2 THEN 57Ø
58Ø IF TR=2 AND L2+(RL1+RL2)=L1 THEN 57Ø
581 IF TR=1 AND L1+(RL1+RL2)=3Ø AND (L2=1Ø) THEN 57Ø
582 IF TR=2 AND L2+(RL1+RL2)=3Ø AND (L1=1Ø) THEN 57Ø
583 IF (TR=1 AND L1=4Ø) AND (RL1+RL2)=L2 THEN 57Ø
584 IF (TR=2 AND L2=4Ø) AND (RL1+RL2)=L1 THEN 57Ø
585 IF RL1=RL2 THEN DB=1
59Ø LOCATE 21,32:COLOR 1:PRINT"HIT A KEY TO STOP";
595 RX=INT(RND(1)*6)+1:RY=INT(RND(1)*6)+1
6ØØ IF RX=1 THEN PUT(258,14Ø),NU,PSET ELSE IF RX=2 THEN PUT(258,14Ø),NV,PSET ELS
E IF RX=3 THEN PUT(258,14Ø),NW,PSET ELSE IF RX=4 THEN PUT(258,14Ø),NX,PSET ELSE
IF RX=5 THEN PUT(258,14Ø),NY,PSET ELSE IF RX=6 THEN PUT(258,14Ø),NZ,PSET
6Ø5 SOUND INT(RND(1)*3ØØ),.1
61Ø IF RY=1 THEN PUT(347,14Ø),NU,PSET ELSE IF RY=2 THEN PUT(347,14Ø),NV,PSET ELS
E IF RY=3 THEN PUT(347,14Ø),NW,PSET ELSE IF RY=4 THEN PUT(347,14Ø),NX,PSET ELSE
IF RY=5 THEN PUT(347,14Ø),NY,PSET ELSE PUT(347,14Ø),NZ,PSET
615 AK$=INKEY$:IF AK$=""THEN 595
62Ø LOCATE 21,32:PRINT STRING$(17," ");
625 IF RL1=1 THEN PUT(258,14Ø),NU,PSET ELSE IF RL1=2 THEN PUT(258,14Ø),NV,PSET E
LSE IF RL1=3 THEN PUT(258,14Ø),NW,PSET ELSE IF RL1=4 THEN PUT(258,14Ø),NX,PSET E
LSE IF RL1=5 THEN PUT(258,14Ø),NY,PSET ELSE PUT(258,14Ø),NZ,PSET
63Ø IF RL2=1 THEN PUT(347,14Ø),NU,PSET ELSE IF RL2=2 THEN PUT(347,14Ø),NV,PSET E
LSE IF RL2=3 THEN PUT(347,14Ø),NW,PSET ELSE IF RL2=4 THEN PUT(347,14Ø),NX,PSET E
LSE IF RL2=5 THEN PUT(347,14Ø),NY,PSET ELSE PUT(347,14Ø),NZ,PSET
```

```
635 LOCATE 12,17:PRINT "YOU ROLLED A ";RL1;" AND ";RL2;" FOR A MOVE OF ";RL1+RL2
64Ø IF TR=1 THEN GOSUB 2665 ELSE GOSUB 274Ø
645 RETURN
65Ø '
655 '** WIPE CLEAN BOX 1 **
66Ø '
665 FOR WIPE=6 TO 8:LOCATE WIPE,17:PRINT STRING$(47," ");:NEXT
67Ø RETURN
675 '
68Ø '** WIPE CLEAN BOX 2 **
685 '
69Ø FOR WIPE=11 TO 13:LOCATE WIPE,17:PRINT STRING$(47," ");:NEXT
695 RETURN
7ØØ '
7Ø5 '
71Ø '**** MAIN MENU HERE ****
715 '
72Ø GOSUB 86Ø:FOR DL=1 TO 12ØØ:NEXT:GOSUB 69Ø:LOCATE 11,17:PRINT N$(TR);"'s Choi
ce? -->";:LOCATE 7,23:PRINT M(1);:LOCATE 7,53:PRINT M(2);
725 IF M(TR)+PRP(TR)=<Ø THEN 233Ø
73Ø IF M(TR)<1 THEN LOCATE 11,4Ø:PRINT"[ YOU MUST MORTGAGE ! ]";:COLOR 2:LOCATE
12,42:PRINT"TANYOPOLY RULE:MUST";:LOCATE 13,42:PRINT"KEEP CASH ON HAND !";:COLOR
 1:FOR DL=1 TO 1ØØØ:NEXT:GOTO 1255
735 LOCATE 11,4Ø:COLOR 1:IF TR=1 THEN PRINT"[ ";P$(L1);" ]";:ELSE PRINT "[ ";P$(
L2);" ]";
74Ø LOCATE 12,22:PRINT"<B>uy     <A>ssets   <I>mprove   <T>rade"
745 LOCATE 13,22:PRINT"<M>ort    <U>nmort   <S>ummary   <N>ext";
75Ø AK$=INKEY$:IF AK$=""THEN 75Ø
755 IF AK$="B"OR AK$="b" THEN 795 ELSE IF AK$="I"OR AK$="i" THEN 92Ø ELSE IF AK$
="t"OR AK$="T"THEN 198Ø ELSE IF AK$="S" OR AK$="s" THEN 286Ø ELSE 76Ø
76Ø IF AK$="A"OR AK$="a"THEN 113Ø ELSE IF AK$="M"OR AK$="m"THEN 1255 ELSE IF AK$
="U"OR AK$="u"THEN 138Ø ELSE 765
765 IF AK$="N" OR AK$="n"THEN 54Ø
77Ø BEEP:LOCATE 13,49:GOTO 75Ø
775 '
78Ø '
785 '*** BUY PROPERTY HERE ***
79Ø '
795 GOSUB 69Ø:LOCATE 11,17:PRINT"LOCATION:";:IF TR=1 THEN PRINT P$(L1) ELSE PRIN
T P$(L2)
8ØØ IF TR=1 AND O(L1)=5 THEN LOCATE 12,17:COLOR 2:PRINT"You can't BUY that!":BEE
P:COLOR 1:GOTO 72Ø
8Ø5 IF TR=2 AND O(L2)=5 THEN LOCATE 12,17:COLOR 2:PRINT"You can't BUY that!":BEE
P:COLOR 1:GOTO 72Ø
81Ø IF TR=1 AND O(L1)=Ø THEN LOCATE 12,17:COLOR 1:PRINT"PURCHASE PRICE=$";P(L1):
COLOR 1:M(1)=M(1)-P(L1):PRP(1)=PRP(1)+MV(L1):LOCATE 7,23:PRINT M(1);"   ";:O(L1)
=1:IF L1=5 OR L1=15 OR L1=25 OR L1=35 THEN RROWN(1)=RROWN(1)+1:GOTO 72Ø:ELSE GOT
O 72Ø
815 IF TR=2 AND O(L2)=Ø THEN LOCATE 12,17:COLOR 1:PRINT"PURCHASE PRICE=$";P(L2):
M(2)=M(2)-P(L2):PRP(2)=PRP(2)+MV(L2):LOCATE 7,53:PRINT M(2);"   ";:O(L2)=2:IF L2
=5 OR L2=15 OR L2=25 OR L2=35 THEN RROWN(2)=RROWN(2)+1:GOTO 72Ø:ELSE GOTO 72Ø
82Ø IF TR=1 AND O(L1)=2 OR O(L1)=4 THEN COLOR 2:LOCATE 12,17:PRINT N$(2);" OWNS
THAT!!!":BEEP:COLOR 1:GOTO 72Ø
825 IF TR=2 AND O(L2)=1 OR O(L2)=3 THEN COLOR 2:LOCATE 12,17:PRINT N$(1);" OWNS
THAT!!!":BEEP:COLOR 1:GOTO 72Ø
83Ø IF TR=1 AND O(L1)=1 OR O(L1)=3 THEN LOCATE 12,17:COLOR 1:PRINT"It's Already
YOURS!"
835 IF TR=2 AND O(L2)=2 OR O(L2)=4 THEN LOCATE 12,17:COLOR 1:PRINT"It's Already
YOURS!"
```

```
84Ø FOR DL=1 TO 9ØØ:NEXT:GOTO 72Ø
845 '
85Ø *** CHECK FOR BOOK STATUS HERE ***
855 '
86Ø IF (O(1)=TR OR O(1)=TR+2) AND (O(3)=TR OR O(3)=TR+2)THEN B(1)=TR:B(3)=TR
865 IF (O(6)=TR OR O(6)=TR+2) AND (O(8)=TR OR O(8)=TR+2) AND (O(9)=TR OR O(9)=TR
+2) THEN B(6)=TR:B(8)=TR:B(9)=TR
87Ø IF (O(11)=TR OR O(11)=TR+2) AND (O(13)=TR OR O(13)=TR+2) AND (O(14)=TR OR O(
14)=TR+2) THEN B(11)=TR:B(13)=TR:B(14)=TR
875 IF (O(16)=TR OR O(16)=TR+2) AND (O(18)=TR OR O(18)=TR+2) AND (O(19)=TR OR O(
19)=TR+2) THEN B(16)=TR:B(18)=TR:B(19)=TR
88Ø IF (O(21)=TR OR O(21)=TR+2) AND (O(23)=TR OR O(23)=TR+2) AND (O(24)=TR OR O(
24)=TR+2) THEN B(21)=TR:B(23)=TR:B(24)=TR
885 IF (O(26)=TR OR O(26)=TR+2) AND (O(27)=TR OR O(27)=TR+2) AND (O(29)=TR OR O(
29)=TR+2) THEN B(26)=TR:B(27)=TR:B(29)=TR
89Ø IF (O(31)=TR OR O(31)=TR+2) AND (O(32)=TR OR O(32)=TR+2) AND (O(34)=TR OR O(
34)=TR+2) THEN B(31)=TR:B(32)=TR:B(34)=TR
895 IF (O(37)=TR OR O(37)=TR+2) AND (O(39)=TR OR O(39)=TR+2) THEN B(37)=TR:B(39)
=TR
9ØØ RETURN
9Ø5 '
91Ø '***** IMPROVE PROPERTY GOES HERE *****
915 '
92Ø GOSUB 69Ø:LOCATE 11,31:COLOR 1:PRINT"WHICH TO IMPROVE ?";:COLOR 3:LOCATE 12,
19:PRINT"THIS ONE?";:COLOR 2:LOCATE 13,2Ø:PRINT"[ Y/N ]";:COLOR 1
925 FOR PROP= 1 TO 4Ø:LOCATE 13,56:PRINT PROP;
93Ø GOTO 1Ø65
935 AK$=INKEY$:IF AK$=""THEN 935
94Ø IF AK$="y" OR AK$="Y" THEN IM=PROP:LOCATE 12,31:PRINT STRING$(18," ");:GOTO
96Ø
945 LOCATE 12,31:PRINT STRING$(18," ");
95Ø NEXT
955 GOTO 72Ø
96Ø GOSUB 69Ø:LOCATE 11,17:PRINT"HOUSING CONSTRUCTION:":LOCATE 12,17:COLOR 2:PRI
NT"MUST BUY 1 FOR EACH";:LOCATE 13,17:PRINT"PIECE OF THE BOOK !":COLOR 1
965 IF IM=1 OR IM=3 OR IM=37 OR IM=39 THEN MINHS=2 ELSE MINHS=3
97Ø LOCATE 11,42:PRINT P$(IM);:LOCATE 12,42:PRINT"Min.=";MINHS;"* $"HC(IM);
975 LOCATE 13,42:COLOR 1:PRINT"PURCHASE [Y/N]?     ";:COLOR 1
98Ø AK$=INKEY$:IF AK$=""THEN 98Ø ELSE IF AK$="N"OR AK$="n"THEN 72Ø
985 IF AK$="Y" OR AK$="y" THEN 995
99Ø BEEP:GOTO 98Ø
995 IF M(TR) < (MINHS*HC(IM)) THEN BEEP:BEEP:GOTO 72Ø
1ØØØ M(TR)=M(TR)-(MINHS*HC(IM))
1ØØ5 LOCATE 7,23:PRINT M(1);:LOCATE 7,53:PRINT M(2);
1Ø1Ø IF IM=1 OR IM=3 THEN H(1)=H(1)+1:H(3)=H(3)+1
1Ø15 IF IM=6 OR IM=8 OR IM=9 THEN H(6)=H(6)+1:H(8)=H(8)+1:H(9)=H(9)+1
1Ø2Ø IF IM=11 OR IM=13 OR IM=14 THEN H(11)=H(11)+1:H(13)=H(13)+1:H(14)=H(14)+1
1Ø25 IF IM=16 OR IM=18 OR IM=19 THEN H(16)=H(16)+1:H(18)=H(18)+1:H(19)=H(19)+1
1Ø3Ø IF IM=21 OR IM=23 OR IM=24 THEN H(21)=H(21)+1:H(23)=H(23)+1:H(24)=H(24)+1
1Ø35 IF IM=26 OR IM=27 OR IM=29 THEN H(26)=H(26)+1:H(27)=H(27)+1:H(29)=H(29)+1
1Ø4Ø IF IM=31 OR IM=32 OR IM=34 THEN H(31)=H(31)+1:H(32)=H(32)+1:H(34)=H(34)+1
1Ø45 IF IM=37 OR IM=39 THEN H(37)=H(37)+1:H(39)=H(39)+1
1Ø5Ø IF TR=1 THEN PRP(1)=PRP(1)+(MINHS*HC(IM))/2:TH(1)=TH(1)+MINHS
1Ø55 IF TR=2 THEN PRP(2)=PRP(2)+(MINHS*HC(IM))/2:TH(2)=TH(2)+MINHS
1Ø6Ø GOTO 72Ø
1Ø65 IF B(PROP)<>TR THEN 945
1Ø7Ø IF H(PROP)=5 THEN 945
1Ø75 IF (PROP=6 OR PROP=8 OR PROP=9) AND (O(6)<>TR OR O(8)<>TR OR O(9)<>TR) THEN
 945
```

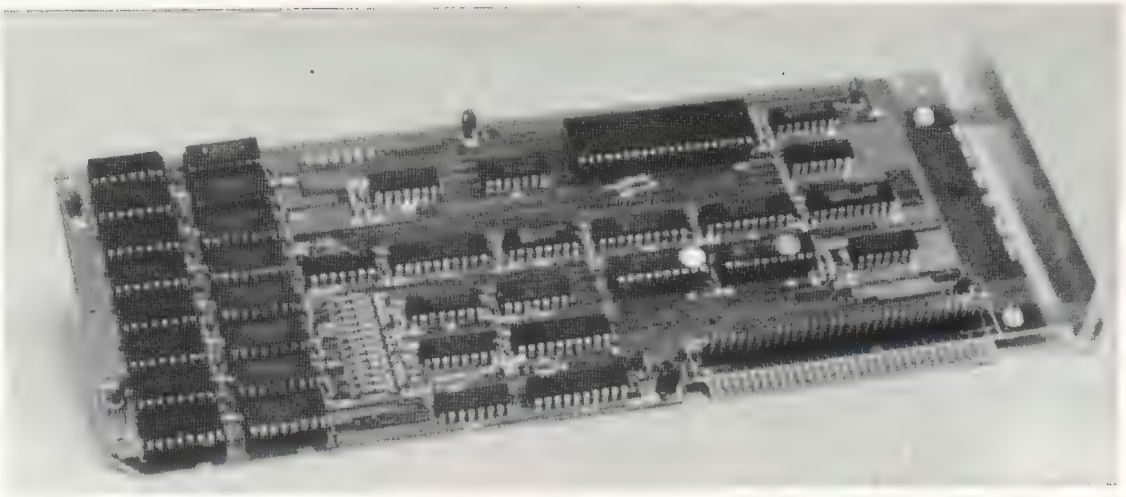
```
1Ø8Ø IF (PROP=11 OR PROP=13 OR PROP=14) AND (O(11)<>TR OR O(13)<>TR OR O(14)<>TR
) THEN 945
1Ø85 IF (PROP=16 OR PROP=18 OR PROP=19) AND (O(16)<>TR OR O(18)<>TR OR O(19)<>TR
) THEN 945
1Ø9Ø IF (PROP=21 OR PROP=23 OR PROP=24) AND (O(21)<>TR OR O(23)<>TR OR O(24)<>TR
) THEN 945
1Ø95 IF (PROP=26 OR PROP=27 OR PROP=29) AND (O(26)<>TR OR O(27)<>TR OR O(29)<>TR
) THEN 945
11ØØ IF (PROP=31 OR PROP=32 OR PROP=34) AND (O(31)<>TR OR O(32)<>TR OR O(34)<>TR
) THEN 945
11Ø5 IF (PROP=37 OR PROP=39) AND (O(37)<>TR OR O(39)<>TR) THEN 945
111Ø LOCATE 12,31:PRINT P$(PROP);:GOTO 935
1115 '
112Ø '***** ASSETS GO HERE *****
1125 '
113Ø GOSUB 69Ø:LOCATE 8,17:PRINT STRING$(45," ");:LOCATE 8,17:COLOR 1:PRINT"ASSE
TS OF:";:COLOR 2:PRINT N$(TR);"   ";:COLOR 1:PRINT"--> <P>rinter   <S>creen";:BE
EP:ASSET(1)=M(1)+PRP(1):ASSET(2)=M(2)+PRP(2)
1135 LOCATE 8,61:AK$=INKEY$:IF AK$=""THEN 1135
114Ø IF AK$="P" OR AK$="p" THEN 1185 ELSE 1145
1145 LOCATE 11,17:PRINT"ASSETS(Cash & Prop)";:LOCATE 12,17:COLOR 2:PRINT"HIT KEY
 TO CONT->";:COLOR 1:LOCATE 13,17:PRINT"TOTAL $";ASSET(TR);
115Ø FOR PROP=1 TO 4Ø:LOCATE 13,56:PRINT PROP;:IF O(PROP)=TR THEN 1155 ELSE 117Ø
1155 FOR WIPE=11 TO 13:LOCATE WIPE,36:PRINT STRING$(28," ");:NEXT
116Ø LOCATE 11,36:PRINT P$(PROP);:LOCATE 12,36:PRINT"MORT VALUE $";MV(PROP);:IF
H(PROP)>Ø AND H(PROP)<5 THEN PRINT"HOUSES:";H(PROP);ELSE IF H(PROP)=5 THEN PRINT
 "Hotel!"
1165 AK$=INKEY$:IF AK$=""THEN 1165
117Ø NEXT PROP
1175 FOR WIPE=11 TO 13:LOCATE WIPE,36:PRINT STRING$(28," ");:NEXT:LOCATE 11,36:P
RINT"THAT'S ALL!":PLAY"V15L64O3E"
118Ø LOCATE 8,17:PRINT STRING$(45," ");:GOTO 72Ø
1185 GOSUB 69Ø:PZ$=N$(TR)+" 'S ASSETS":IF LEN(N$(TR))>4 THEN OV=17Ø ELSE OV=2ØØ
119Ø LOCATE 12,2Ø:PRINT PZ$
1195 LPRINT:LPRINT STRING$(4Ø,"*"):LPRINT "TANYOPOLY ASSET PRINTOUT FOR ";N$(TR)
:LPRINT "TOTAL ASSETS:$";ASSET(TR):LPRINT
12ØØ FOR PROP=1 TO 4Ø:IF O(PROP)=TR THEN 12Ø5 ELSE 123Ø
12Ø5 LPRINT P$(PROP);:LPRINT TAB(18);" MORTGAGE VALUE: $"MV(PROP)
121Ø IF B(PROP)=TR THEN LPRINT"You Own The Book."
1215 IF H(PROP)>Ø AND H(PROP)<5 THEN LPRINT "HOUSES BUILT: ";H(PROP);ELSE IF H(P
ROP)=5 THEN PRINT"Hotel!"
122Ø IF O(PROP)=3 OR O(PROP)=4 THEN LPRINT"This Property Mortgaged!"
1225 LPRINT
123Ø NEXT PROP
1235 LPRINT STRING$(4Ø,"*"):LPRINT:LOCATE 8,17:PRINT STRING$(45," "):GOTO 72Ø
124Ø '
1245 '***** MORTGAGE PROPERTY GOES HERE *****
125Ø '
1255 GOSUB 69Ø:LOCATE 11,17:PRINT"MORTGAGING PROPERTY";:LOCATE 12,17:COLOR 2:PRI
NT"MUST MORTAGE HOUSES";:LOCATE 13,17:PRINT"BEFORE THE PROPERTY";:COLOR 1
126Ø PLAY"V15L64O4FDC":FOR DL=1 TO 2ØØØ:NEXT:LOCATE 12,17:PRINT"     THIS ONE  -
-->";:LOCATE 13,17:COLOR 2:PRINT "          [ Y/N ] ? ";:COLOR 1
1265 FOR UPROP=1 TO 4Ø:LOCATE 13,56:PRINT UPROP;
127Ø IF (TR=1 AND O(UPROP)=1) OR (TR=2 AND O(UPROP)=2) THEN LOCATE 13,56:PRINT"
 ";:LOCATE 11,42:PRINT P$(UPROP);:ELSE 1355
1275 LOCATE 12,42:PRINT"MORTGAGE VALUE $"MV(UPROP);
128Ø LOCATE 13,42:PRINT"HOUSES : ";H(UPROP);"     ";
1285 AK$=INKEY$:IF AK$=""THEN 1285 ELSE IF AK$="n" OR AK$="N" THEN 1355 ELSE 129
Ø
```

Tandy 1000 Memory Card

another high quality product from Southwestern Digital

Why spend a bundle on a Tandy board?
Our Board is only

$135.



Features:

- 512K of Memory
- DMA
- Expansion Port
- Gold Edge Cards
- Easy Installation
- 30 Day Satisfaction Guarantee Policy

The Southwestern Digital Memory Expansion Plus Card has all the features of the Radio Shack Board but the price; you save almost $400. Features include 512K installed, burned in, and tested to give you a total of 640K, a DMA circut that is fully tested for hard drive operation, and an expansion port that will work with any of the Radio Shack Memory Plus Expansion Card options. High quality manufacturing, and features such as gold plated card edges make this the logical choice in upgrading your memory.

Ordering Information
Call us or mail your order in. We accept Visa, Mastercard, and Certified Funds for quickest shipment. Personal checks are held for clearance. Add $5. for ground shipment, or $10 for UPS 2nd day air service. All products carry a 30 day satisfaction guarantee, and are warrented for a full year.

Order Line
1-713-480-3296
Southwestern Digital
2515 West Main Street, Suite #337
Houston, Texas 77098

```
129Ø IF AK$="Y" OR AK$="y"THEN 1295 ELSE 1285
1295 IF H(UPROP)=Ø THEN 1345
13ØØ IF (UPROP=1 OR UPROP=3) THEN H(1)=H(1)-1:H(3)=H(3)-1:M(TR)=M(TR)+(2*HC(UPRO
P))/2:PRP(TR)=PRP(TR)-(2*HC(UPROP)):TH(TR)=TH(TR)-2
13Ø5 IF (UPROP=6 OR UPROP=8 OR UPROP=9) THEN H(6)=H(6)-1:H(8)=H(8)-1:H(9)=H(9)-1
:M(TR)=M(TR)+(3*HC(UPROP))/2:PRP(TR)=PRP(TR)-(2*HC(UPROP)):TH(TR)=TH(TR)-3
131Ø IF (UPROP=11 OR UPROP=13 OR UPROP=14) THEN H(11)=H(11)-1:H(13)=H(13)-1:H(14
)=H(14)-1:M(TR)=M(TR)+(3*HC(UPROP))/2:PRP(TR)=PRP(TR)-(2*HC(UPROP)):TH(TR)=TH(TR
)-3
1315 IF (UPROP=16 OR UPROP=18 OR UPROP=19) THEN H(16)=H(16)-1:H(18)=H(18)-1:H(19
)=H(19)-1:M(TR)=M(TR)+(3*HC(UPROP))/2:PRP(TR)=PRP(TR)-(2*HC(UPROP)):TH(TR)=TH(TR
)-3
132Ø IF (UPROP=21 OR UPROP=23 OR UPROP=24) THEN H(21)=H(21)-1:H(23)=H(23)-1:H(24
)=H(24)-1:M(TR)=M(TR)+(3*HC(UPROP))/2:PRP(TR)=PRP(TR)-(2*HC(UPROP)):TH(TR)=TH(TR
)-3
1325 IF (UPROP=26 OR UPROP=27 OR UPROP=29) THEN H(26)=H(26)-1:H(27)=H(27)-1:H(29
)=H(29)-1:M(TR)=M(TR)+(3*HC(UPROP))/2:PRP(TR)=PRP(TR)-(2*HC(UPROP)):TH(TR)=TH(TR
)-3
133Ø IF (UPROP=31 OR UPROP=32 OR UPROP=33) THEN H(31)=H(31)-1:H(32)=H(32)-1:H(33
)=H(33)-1:M(TR)=M(TR)+(3*HC(UPROP))/2:PRP(TR)=PRP(TR)-(2*HC(UPROP)):TH(TR)=TH(TR
)-3
1335 IF (UPROP=37 OR UPROP=39) THEN H(37)=H(37)-1:H(39)=H(39)-1:M(TR)=M(TR)+(2*H
C(UPROP))/2:PRP(TR)=PRP(TR)-(2*HC(UPROP)):TH(TR)=TH(TR)-2
134Ø LOCATE 7,23:PRINT M(1);:LOCATE 7,53:PRINT M(2);:GOTO 136Ø
1345 M(TR)=M(TR)+MV(UPROP):PRP(TR)=PRP(TR)-MV(UPROP):IF TR=1 THEN O(UPROP)=3 ELS
E IF TR=2 THEN O(UPROP)=4
135Ø LOCATE 7,23:PRINT M(1);:LOCATE 7,53:PRINT M(2);:GOTO 136Ø
1355 FOR WIPE=11 TO 13:LOCATE WIPE,42:PRINT STRING$(22," "):NEXT:NEXT
136Ø GOTO 72Ø
1365 '
137Ø '***** UNMORTGAGE PROPERTY GOES HERE *****
1375 '
138Ø GOSUB 69Ø:LOCATE 11,17:PRINT"UNMORTGAGE PROPERTY:";:LOCATE 12,27:COLOR 2:PR
INT"THIS ONE->";:LOCATE 13,28:PRINT"[ Y/N ]";:COLOR 1
1385 FOR PROP=1 TO 4Ø:LOCATE 13,56:PRINT PROP;
139Ø IF (TR=1 AND O(PROP)=3) OR (TR=2 AND O(PROP)=4) THEN LOCATE 12,42:PRINT P$(
PROP);:ELSE 1435
1395 LOCATE 13,42:PRINT"COST: $";UN(PROP);"    ";
14ØØ AK$=INKEY$:IF AK$=""THEN 14ØØ
14Ø5 IF AK$="N"OR AK$="n" THEN 1435
141Ø IF AK$="y" OR AK$="Y" THEN 1415 ELSE 14Ø5
1415 M(TR)=M(TR)-UN(PROP):O(PROP)=O(PROP)-2:PRP(TR)=PRP(TR)+MV(PROP)
142Ø LOCATE 7,23:PRINT M(1);:LOCATE 7,53:PRINT M(2);
1425 FOR WIPE=12 TO 13:LOCATE WIPE,4Ø:PRINT STRING$(24," ");:NEXT
143Ø PLAY"V12L64O5CO2C":GOTO 72Ø
1435 FOR WIPE=12 TO 13:LOCATE WIPE,4Ø:PRINT STRING$(24," ");:NEXT:NEXT
144Ø LOCATE 12,4Ø:PRINT"THAT'S EVERYTHING!";:PLAY"V15L64O2CO5C"
1445 GOTO 72Ø
145Ø '
1455 '
146Ø '***** RENT DUE!! *****
1465 '
147Ø IF TR=2 THEN 15ØØ ELSE GOSUB 69Ø:LOCATE 11,17:COLOR 2,Ø:PRINT"RENT DUE  :";
:COLOR 1:PRINT" on ";P$(L1);
1475 IF TR=1 THEN IF H(L1)=Ø THEN RD=RØ(L1) ELSE IF H(L1)=1 THEN RD=R1(L1) ELSE
IF H(L1)=2 THEN RD=R2(L1) ELSE IF H(L1)=3 THEN RD=R3(L1) ELSE IF H(L1)=4 THEN RD
=R4(L1) ELSE IF H(L1)=5 THEN RD=R5(L1)
148Ø IF B(L1)=2 AND H(L1)=Ø THEN RD=RD*2
1485 LOCATE 12,17:COLOR 1:PRINT"PAY $";RD:IF B(L1)=2 THEN LOCATE 12,27:PRINT"[BO
```

```
OK]";
149Ø IF M(TR)+PRP(TR)<RD THEN FOR DL=1 TO 9ØØ:NEXT:GOTO 233Ø
1495 M(1)=M(1)-RD:LOCATE 7,23:PRINT"     ";:FOR RT=1 TO 4:COLOR 1:LOCATE 7,23:PR
INT M(1);:PLAY"V15L64O2D":COLOR 2:NEXT:COLOR 1:LOCATE 7,23:PRINT M(1);:FOR DL=1
TO 9ØØ:NEXT:M(2)=M(2)+RD:LOCATE 7,53:PRINT"     ";:LOCATE 7,53:PRINT M(2);:RETUR
N
15ØØ GOSUB 69Ø:LOCATE 11,17:COLOR 2,Ø:PRINT"RENT DUE  :";:COLOR 1:PRINT" on ";P$
(L2);
15Ø5 IF TR=2 THEN IF H(L2)=Ø THEN RD=RØ(L2) ELSE IF H(L2)=1 THEN RD=R1(L2) ELSE
IF H(L2)=2 THEN RD=R2(L2) ELSE IF H(L2)=3 THEN RD=R3(L2) ELSE IF H(L2)=4 THEN RD
=R4(L2) ELSE IF H(L2)=5 THEN RD=R5(L2)
151Ø IF B(L2)=1 AND H(L2)=Ø THEN RD=RD*2
1515 LOCATE 12,17:COLOR 1:PRINT"PAY $";RD:IF B(L2)=1 THEN LOCATE 12,27:PRINT"[BO
OK]";
152Ø IF M(2)+PRP(2)<RD THEN FOR DL=1 TO 9ØØ:NEXT:GOTO 2355
1525 M(2)=M(2)-RD:LOCATE 7,53:PRINT"     ";:FOR RT=1 TO 4:COLOR 1:LOCATE 7,53:PR
INT M(2);:PLAY"V15L64O2D":COLOR 2:NEXT:COLOR 1:LOCATE 7,53:PRINT M(2);:FOR DL=1
TO 9ØØ:NEXT:M(1)=M(1)+RD:LOCATE 7,23:PRINT"     ";:LOCATE 7,23:PRINT M(1);:RETUR
N
153Ø '
1535 '***** COUNTY CHEST RESPONSE *****
154Ø '
1545 FOR DL=1 TO 9ØØ:NEXT:PLAY"O4D":GOSUB 69Ø:LOCATE 11,26:COLOR 2:PRINT"C O U N
 T Y __ C H E S T !":COLOR 1
155Ø FOR WIPE=18 TO 19:LOCATE WIPE,52:PRINT STRING$(12," ");:NEXT
1555 CC=INT(RND(1)*12)+1
156Ø ON CC GOTO 1565,157Ø,1575,161Ø,1615,162Ø,1625,163Ø,1635,164Ø,1645,165Ø
1565 IF TR=1 THEN M(2)=M(2)-5Ø:M(1)=M(1)+5Ø:GOTO 1655:ELSE IF TR=2 THEN M(1)=M(1
)-5Ø:M(2)=M(2)+5Ø:GOTO 1655
157Ø M(TR)=M(TR)+45:GOTO 1655
1575 IF (TR=1 AND L2=4Ø) OR (TR=2 AND L1=4Ø) THEN 1555 ELSE IF TR=2 THEN 1595 EL
SE 158Ø
158Ø PUT(O1(L1),D1(L1)),U,PSET:L1=4Ø:GET(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),U
1585 LINE(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),2,BF
159Ø LINE(O1(L1)+1,D1(L1)+1)-(O1(L1)+14,D1(L1)+7),1,B:GOTO 1655
1595 PUT(O1(L2),D1(L2)),W,PSET:L2=4Ø:GET(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),W
16ØØ LINE(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),3,BF
16Ø5 LINE(O1(L2)+1,D1(L2)+1)-(O1(L2)+14,D1(L2)+7),1,B:GOTO 1655
161Ø M(TR)=M(TR)+2ØØ:GOTO 1655
1615 M(TR)=M(TR)+25:GOTO 1655
162Ø IF (TR=1 AND L2=1Ø) OR (TR=2 AND L1=1Ø) THEN 1555 ELSE FF=2:GOTO 1655
1625 M(TR)=M(TR)-1ØØ:FM=FM+1ØØ:GOTO 1655
163Ø M(TR)=M(TR)+2Ø:GOTO 1655
1635 M(TR)=M(TR)-15Ø:FM=FM+1ØØ:GOTO 1655
164Ø M(TR)=M(TR)+1ØØ:GOTO 1655
1645 M(TR)=M(TR)-5Ø:GOTO 1655
165Ø M(TR)=M(TR)+12:GOTO 1655
1655 LOCATE 18,52:PRINT C4$(CC):LOCATE 19,52:PRINT C5$(CC):PLAY"V15L64O5CO3CO1C"
:LOCATE 7,23:PRINT M(1);"   ":LOCATE 7,53:PRINT M(2);"   ";:FOR DL=1 TO 28ØØ:NEX
T:IF FF=2 THEN 221Ø
166Ø FOR WIPE=18 TO 19:LOCATE WIPE,52:PRINT STRING$(12," ");:NEXT:LOCATE 18,52:P
RINT " C H E S T ":FF=Ø
1665 GOTO 5ØØ
167Ø '
1675 '***** CHANCE RESPONSE *****
168Ø '
1685 FOR DL=1 TO 9ØØ:NEXT:PLAY"V1ØL64T255O3CO4CO5C":GOSUB 69Ø:LOCATE 11,34:COLOR
 1:PRINT"C H A N C E !"
169Ø FOR WIPE=18 TO 19:LOCATE WIPE,17:PRINT STRING$(13," ");:NEXT
```

```
1695 CC=INT(RND(1)*12)+1
17ØØ ON CC GOTO 17Ø5,171Ø,1715,172Ø,179Ø,1795,183Ø,1865,19ØØ,19Ø5,194Ø,1945
17Ø5 M(TR)=M(TR)-25:GOTO 195Ø
171Ø FINE=25*TH(TR):IF FINE=>(M(TR)+PRP(TR)) THEN 233Ø ELSE M(TR)=M(TR)-FINE:FM=
FM+FINE:GOTO 195Ø
1715 JF(TR)=1:GOTO 195Ø
172Ø IF TR=2 THEN 1755 ELSE IF (L1<5 AND L2=5) OR (L1>4 AND L1<15 AND L2=15) OR
(L1>14 AND L1<25 AND L2=25) OR (L1>25 AND L2=35) THEN 1695
1725 PUT(O1(L1),D1(L1)),U,PSET
173Ø IF L1<5 THEN L1=5 ELSE IF (L1>4 AND L1<15) THEN L1=15 ELSE IF (L1>14 AND L1
<25) THEN L1=25 ELSE IF (L1>24 AND L1<35) THEN L1=35 ELSE IF (L1>35 AND L1<4Ø) T
HEN L1=5
1735 GET(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),U
174Ø LINE(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),2,BF
1745 LINE(O1(L1)+1,D1(L1)+1)-(O1(L1)+14,D1(L1)+7),1,B
175Ø GOTO 195Ø
1755 IF (L2<5 AND L1=5) OR (L2>4 AND L2<15 AND L1=15) OR (L2>14 AND L2<25 AND L1
=25) OR (L2>25 AND L1=35) THEN 1695
176Ø PUT(O1(L2),D1(L2)),W,PSET
1765 IF L2<5 THEN L2=5 ELSE IF (L2>4 AND L2<15) THEN L2=15 ELSE IF (L2>14 AND L2
<25) THEN L2=25 ELSE IF (L2>24 AND L2<35) THEN L2=35 ELSE IF (L2>35 AND L2<4Ø) T
HEN L2=5
177Ø GET(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),W
1775 LINE(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),3,BF
178Ø LINE(O1(L2)+1,D1(L2)+1)-(O1(L2)+14,D1(L2)+7),1,B
1785 GOTO 195Ø
179Ø IF TR=1 THEN M(1)=M(1)-5Ø:M(2)=M(2)+5Ø:GOTO 195Ø:ELSE M(2)=M(2)-5Ø: .(1)=M(1
)+5Ø:GOTO 195Ø
1795 IF (TR=1 AND L2=11) OR (TR=2 AND L1=11) THEN 1695 ELSE IF TR=2 THEN 1815 EL
SE 18ØØ
18ØØ PUT(O1(L1),D1(L1)),U,PSET:L1=11:GET(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),U
18Ø5 LINE(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),2,BF
181Ø LINE(O1(L1)+1,D1(L1)+1)-(O1(L1)+14,D1(L1)+7),1,B:GOTO 195Ø
1815 PUT(O1(L2),D1(L2)),W,PSET:L2=11:GET(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),W
182Ø LINE(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),3,BF
1825 LINE(O1(L2)+1,D1(L2)+1)-(O1(L2)+14,D1(L2)+7),1,B:GOTO 195Ø
183Ø IF (TR=1 AND L2=4Ø) OR (TR=2 AND L1=4Ø) THEN 1695 ELSE IF TR=2 THEN 185Ø EL
SE 1835
1835 PUT(O1(L1),D1(L1)),U,PSET:L1=4Ø:GET(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),U
184Ø LINE(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),2,BF
1845 LINE(O1(L1)+1,D1(L1)+1)-(O1(L1)+14,D1(L1)+7),1,B:GOTO 195Ø
185Ø PUT(O1(L2),D1(L2)),W,PSET:L2=4Ø:GET(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),W
1855 LINE(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),3,BF
186Ø LINE(O1(L2)+1,D1(L2)+1)-(O1(L2)+14,D1(L2)+7),1,B:GOTO 195Ø
1865 IF (TR=1 AND L2=24) OR (TR=2 AND L1=24) THEN 1695 ELSE IF TR=2 THEN 1885 EL
SE 187Ø
187Ø PUT(O1(L1),D1(L1)),U,PSET:L1=24:GET(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),U
1875 LINE(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),2,BF
188Ø LINE(O1(L1)+1,D1(L1)+1)-(O1(L1)+14,D1(L1)+7),1,B:GOTO 195Ø
1885 PUT(O1(L2),D1(L2)),W,PSET:L2=24:GET(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),W
189Ø LINE(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),3,BF
1895 LINE(O1(L2)+1,D1(L2)+1)-(O1(L2)+14,D1(L2)+7),1,B:GOTO 195Ø
19ØØ FF=1:GOTO 195Ø
19Ø5 IF (TR=1 AND L2=39) OR (TR=2 AND L1=39) THEN 1695 ELSE IF TR=2 THEN 1925 EL
SE 191Ø
191Ø PUT(O1(L1),D1(L1)),U,PSET:L1=39:GET(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),U
1915 LINE(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),2,BF
192Ø LINE(O1(L1)+1,D1(L1)+1)-(O1(L1)+14,D1(L1)+7),1,B:GOTO 195Ø
1925 PUT(O1(L2),D1(L2)),W,PSET:L2=39:GET(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),W
```

```
193Ø LINE(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),3,BF
1935 LINE(O1(L2)+1,D1(L2)+1)-(O1(L2)+14,D1(L2)+7),1,B:GOTO 195Ø
194Ø M(TR)=M(TR)+5Ø:GOTO 195Ø
1945 M(TR)=M(TR)-15:FM=FM+15
195Ø LOCATE 18,18:PRINT C1$(CC):LOCATE 19,18:PRINT C2$(CC):PLAY"V12L64O3C":LOCAT
E 7,23:PRINT M(1);"     ";:LOCATE 7,53:PRINT M(2);"     ";:FOR DL=1 TO 28ØØ:NEXT
:IF FF=1 THEN 221Ø
1955 FOR WIPE=18 TO 19:LOCATE WIPE,17:PRINT STRING$(13," ");:NEXT:LOCATE 18,17:P
RINT "C H A N C E"
196Ø GOTO 5ØØ
1965 '
197Ø '*****  TRADE PROPERTY *****
1975 '
198Ø GOSUB 69Ø:LOCATE 11,17:COLOR 1:PRINT"TRADING ROUTINE: ONLY PROPERTY MAY BE
TRADED!";:LOCATE 12,17:COLOR 1:PRINT N$(1);:LOCATE 12,24:PRINT": THIS ONE (Y/N)?
";
1985 FOR TT=1 TO 4Ø:IF O(TT)=1 AND H(TT)=Ø THEN LOCATE 12,43:PRINT P$(TT);:ELSE
2ØØØ
199Ø AK$=INKEY$:IF AK$=""THEN 199Ø
1995 IF AK$="Y"OR AK$="y" THEN 2Ø1Ø ELSE 2ØØØ
2ØØØ LOCATE 12,43:PRINT STRING$(2Ø," ");:NEXT
2ØØ5 LOCATE 12,43:PRINT"NOTHING TO TRADE!":FOR DL=1 TO 12ØØ:NEXT:GOTO 72Ø
2Ø1Ø LOCATE 13,17:PRINT N$(2);:LOCATE 13,24:PRINT": THIS ONE (Y/N)?";
2Ø15 FOR TU=1 TO 4Ø:IF O(TU)=2 AND H(TU)=Ø THEN LOCATE 13,43:PRINT P$(TU);:ELSE
2Ø35
2Ø2Ø AK$=INKEY$:IF AK$=""THEN 2Ø2Ø
2Ø25 IF AK$="Y" OR AK$="y" THEN 2Ø45 ELSE 2Ø35
2Ø3Ø GOTO 2Ø35
2Ø35 LOCATE 13,43:PRINT STRING$(2Ø," ");:NEXT
2Ø4Ø LOCATE 13,43:PRINT"NOTHING TO TRADE!":FOR DL=1 TO 12ØØ:NEXT:GOTO 72Ø
2Ø45 GOTO 2Ø75
2Ø5Ø O(TT)=2:O(TU)=1:PRP(1)=PRP(1)-MV(TT):PRP(1)=PRP(1)+MV(TU):PRP(2)=PRP(2)-MV(
TU):PRP(2)=PRP(2)+MV(TT)
2Ø55 GOSUB 69Ø:LOCATE 11,17:COLOR 1:PRINT"THE FOLLOWING TRADE IS RECORDED:";
2Ø6Ø LOCATE 12,17:PRINT N$(1);" Trades ";P$(TT);" To"
2Ø65 LOCATE 13,17:PRINT N$(2);" For ";P$(TU);
2Ø7Ø FOR DL=1 TO 12ØØ:NEXT:GOTO 72Ø
2Ø75 GOSUB 69Ø:LOCATE 11,17:PRINT"Closing the deal-";:LOCATE 12,17:PRINT"Trade "
;P$(TT);" for ";P$(TU);
2Ø8Ø LOCATE 13,17:PRINT N$(1);" Input Your Approval (Y/N) ?";:INPUT AP$:IF AP$="
y" OR AP$="Y" THEN 2Ø85 ELSE 72Ø
2Ø85 LOCATE 13,17:PRINT STRING$(32," ");:LOCATE 13,17:PRINT N$(2);" Input Your A
pproval (Y/N) ?";:INPUT AQ$:IF AQ$="y" OR AP$="Y" THEN 2Ø9Ø ELSE 72Ø
2Ø9Ø GOSUB 86Ø
2Ø95 FOR DL=1 TO 8ØØ:NEXT:GOTO 2Ø5Ø
21ØØ '
21Ø5 '***** INCOME TAX *****
211Ø '
2115 GOSUB 69Ø:LOCATE 11,17:COLOR 1:PRINT"INCOME TAX !.......";:COLOR 2:PRINT"No
 EXCUSES!....Pay Up!";:COLOR 1
212Ø LOCATE 13,17:PRINT"Taxes Due...$2ØØ";
2125 FOR DUM=1 TO 2:PLAY"T12ØV15O1L4CP8L8DL6E-L4CP4":NEXT:PLAY"L3F+"
213Ø IF TR=1 THEN M(1)=M(1)-2ØØ ELSE IF TR=2 THEN M(2)=M(2)-2ØØ
2135 LOCATE 7,23:PRINT M(1):LOCATE 7,53:PRINT M(2)
214Ø FOR DL=1 TO 9ØØ:NEXT
2145 RETURN
215Ø '
2155 '***** LUXURY TAX *****
216Ø '
```

```
2165 GOSUB 69Ø:LOCATE 11,17:COLOR 1:PRINT"LUXURY TAX !......";:LOCATE 12,17:COLO
R 1:PRINT"No EXCUSES!.....Pay Up!";
217Ø LOCATE 13,17:PRINT"The TAX is $75.ØØ";
2175 IF TR=1 THEN M(1)=M(1)-75 ELSE IF TR=2 THEN M(2)=M(2)-75
218Ø LOCATE 7,23:PRINT M(1):LOCATE 7,53:PRINT M(2)
2185 FOR DL=1 TO 9ØØ:NEXT
219Ø RETURN
2195 '
22ØØ '*** IN THE JAILHOUSE ***
22Ø5 '
221Ø LOCATE 13,17:PRINT"YOU MOVE TO ";P$(1Ø);STRING$(8," ");:FOR DL=1 TO 1ØØØ:NE
XT
2215 FOR SIREN=1 TO 2:FOR SN=7ØØ TO 1ØØØ STEP 5:SOUND SN,.8:NEXT
222Ø IF SIREN =2 THEN FOR SN=99Ø TO 4ØØ STEP -16:SOUND SN,.4:NEXT
2225 IF SIREN =1 THEN FOR SN=99Ø TO 7ØØ STEP -16:SOUND SN,.8:NEXT
223Ø NEXT
2235 IF TR=1 THEN PUT(O1(L1),D1(L1)),U,PSET:L1=1Ø:GET(O1(1Ø),D1(1Ø))-(O1(1Ø)+15,
D1(1Ø)+8),U:LINE(O1(1Ø),D1(1Ø))-(O1(1Ø)+15,D1(1Ø)+8),2,BF
224Ø IF TR=1 THEN LINE(O1(L1)+1,D1(L1)+1)-(O1(L1)+14,D1(L1)+7),1,B
2245 IF TR=2 THEN PUT(O1(L2),D1(L2)),W,PSET:L2=1Ø:GET(O1(1Ø),D1(1Ø))-(O1(1Ø)+15,
D1(1Ø)+8),W:LINE(O1(1Ø),D1(1Ø))-(O1(1Ø)+15,D1(1Ø)+8),3,BF
225Ø IF TR=2 THEN LINE(O1(L2)+1,D1(L2)+1)-(O1(L2)+14,D1(L2)+7),1,B
2255 IF JF(TR)=Ø THEN LOCATE 21,34:PRINT"PAY $5Ø BOND";:M(TR)=M(TR)-5Ø
226Ø IF JF(TR)=1 THEN LOCATE 21,34:PRINT"GET OUT FREE";:JF(TR)=Ø
2265 FOR DL=1 TO 12ØØ:NEXT:PLAY"V15O3F":LOCATE 21,33:PRINT STRING$(14," ");
227Ø IF FF=2 THEN GOTO 166Ø
2275 IF FF=1 THEN 1955
228Ø GOTO 72Ø
2285 '
229Ø '***** FREE PARKING HERE *****
2295 '
23ØØ LOCATE 21,33:PRINT"YOU WIN $";FM:M(TR)=M(TR)+FM:FM=1ØØ:LOCATE 7,23:PRINT M(
1):LOCATE 7,53:PRINT M(2);
23Ø5 FOR TOOT=1 TO 2:FOR SN=1 TO 35:SOUND 3ØØ,.3:NEXT:FOR DL=1 TO 5Ø:NEXT:NEXT
231Ø FOR DL=1 TO 8ØØ:NEXT:LOCATE 21,33:PRINT STRING$(14," ");:GOTO 72Ø
2315 '
232Ø '***BANKRUPT PLAYER***
2325 '
233Ø GOSUB 69Ø:LOCATE 11,17:COLOR 1:PRINT"BANKRUPCY COURT:";:LOCATE 12,17:COLOR
1:PRINT "YOUR CASH & PROPERTY VALUES ARE INSUFFICIENT!";:LOCATE 13,17:PRINT"IN O
THER WORDS, "N$(TR)" YOU'RE BANKRUPT !":FOR DL=1 TO 25ØØ:NEXT DL
2335 PLAY"V15L64O3CEGO4CO3GEC":EG$="WINNER: "
234Ø IF TR=1 THEN NM$=N$(TR+1) ELSE NM$=N$(1)
2345 GOSUB 69Ø:EG$=EG$+NM$:LOCATE 12,2Ø:PRINT EG$
235Ø GOTO 235Ø
2355 '
236Ø '
2365 'RAILROAD FEES GO HERE
237Ø '
2375 FOR TRAIN=1 TO 2:SOUND 3ØØ,.4:SOUND 5ØØ,.4
238Ø IF TRAIN=1 THEN FOR SN=1 TO 8Ø:SOUND 6ØØ,.1:NEXT
2385 IF TRAIN=2 THEN FOR SN=1 TO 15Ø:SOUND 6ØØ,.1:NEXT
239Ø NEXT TRAIN
2395 IF TR=1 THEN IF RROWN(2)=1 THEN RRIDE=25 ELSE IF RROWN(2)=2 THEN RRIDE=5Ø E
LSE IF RROWN(2)=3 THEN RRIDE=1ØØ ELSE IF RROWN(2)=4 THEN RRIDE=2ØØ
24ØØ IF TR=2 THEN IF RROWN(1)=1 THEN RRIDE=25 ELSE IF RROWN(1)=2 THEN RRIDE=5Ø E
LSE IF RROWN(1)=3 THEN RRIDE=1ØØ ELSE IF RROWN(1)=4 THEN RRIDE=2ØØ
24Ø5 GOSUB 69Ø
241Ø IF TR=1 THEN LOCATE 11,17:PRINT N$(2);" OWNS ";P$(L1);:LOCATE 12,17:PRINT"N
```

Grow up!

Wondering where to grow?

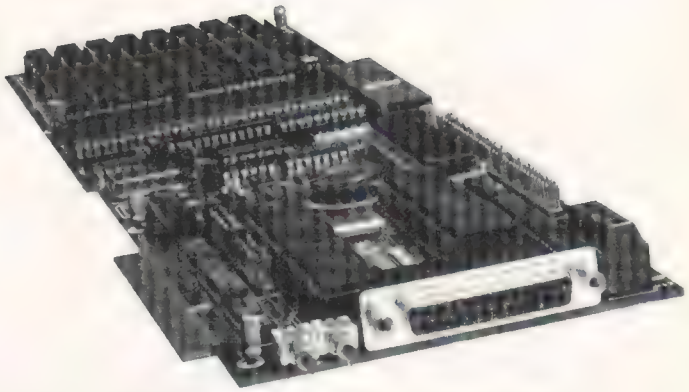
Why not get IBM compatibility with Tandy's 1200, or the new 3000, and take your OS-9 programs with you?

TLM's PC68K puts the 68000/68010 processor and Microware's OS-9 together in any IBM, or compatible, PC/XT/AT system, providing the best of both worlds with fully concurrent PC/68000 operation.

- Powerful languages such as BASIC09, PASCAL and C
- DYNACALC for spread sheet processing
- The power of the 68000/68010
- The price/performance and flexibility of the Tandy 1200 and 3000

TLM's PC68K . . . A great family to grow up with!

IBM is a registered trademark of International Business Machines
DYNACALC is a registered trademark of Computer Systems Center
OS-9 is a registered trademark of Microware Systems Corp
CPM 68K is a registered trademark of Digital Research Corp
MK68000 MK68010 are registered trademarks of Motorola
UNIX is a registered trademark of AT&T



CLOCK SPEED
8/10/12.5 MHZ
Without wait states

HARDWARE
2 Parallel Ports
3-9 Serial Ports
68881 Floating point co-processor (optional)
Memory mapped dual port bus for high speed DMA transfers
Real time clock with battery backup
Up to 16 boards can run concurrently from one AT/XT/PC bus

MEMORY
1-2 MB OF RAM
(1 MB on board standard)
8K-32K of ROM
2K-8K of battery backed RAM

MULTI-USER
3-9 users (3 standard)
High speed "Tick" timer with 10ms accuracy for efficient multi-user time slicing.

OPERATING SYSTEMS
OS-9 (Multi-user UNIX look-alike)
C/PM-68K (popular single user OS)
Both support concurrent access to PC DOS

SOFTWARE
Software selectable OS environment
IBM Color/Mono Graphics Support
Local and Global disk caching for maximum speed and efficient data transfer

COMMUNICATIONS
Powerful asynchronous/synchronous data communications capability
Bisynchronous SNA 3270, X.25, etc.

MISCELLANEOUS FEATURES
Remote and Local reset (hardware or software selectable)
Remote and Local Non-maskable Interrupts (NMI)
Stand alone hardware capability (without AT/XT/PC support)

TLM Systems

West: 4704 W. Jennifer, Suite 105, Fresno, CA 93711 209/276-2345 / East: 67 Grandview Avenue, Pleasantville, NY 10570 914/747-1450

```
UMBER OF RAILROADS OWNED BY ";N$(2);" IS ";RROWN(2);:LOCATE 13,17:PRINT"YOU MUST
 PAY A FEE OF $";RRIDE:M(1)=M(1)-RRIDE:M(2)=M(2)+RRIDE
2415 IF TR=2 THEN LOCATE 11,17:PRINT N$(1);" OWNS ";P$(L2);:LOCATE 12,17:PRINT"N
UMBER OF RAILROADS OWNED BY ";N$(1);" IS ";RROWN(1);:LOCATE 13,17:PRINT"YOU MUST
 PAY A FEE OF $";RRIDE:M(2)=M(2)-RRIDE:M(1)=M(1)+RRIDE
242Ø FOR RT=1 TO 4:COLOR 2:LOCATE 7,23:PRINT M(1);:PLAY"V15L64O2D":COLOR 1:LOCAT
E 7,23:PRINT M(1);:NEXT:LOCATE 7,53:PRINT M(2);:FOR DL=1 TO 9ØØ:NEXT:RETURN
2425 LOCATE 11,17:PRINT N$(1);" OWNS ";P$(L2);:LOCATE 12,17:PRINT"NUMBER OF RAIL
ROADS OWNED BY ";N$(1);" IS ";RROWN(1);:LOCATE 13,17:PRINT"YOU MUST PAY A FEE OF
 $";RRIDE:M(2)=M(2)-RRIDE:M(1)=M(1)+RRIDE
243Ø FOR RT=1 TO 4:COLOR 2:LOCATE 7,53:PRINT M(2);:PLAY"V15L64O2D":COLOR 1:LOCAT
E 7,53:PRINT M(2);:NEXT:LOCATE 7,23:PRINT M(1);:FOR DL=1 TO 9ØØ:NEXT:RETURN
2435 '
244Ø '
2445 '***UTILITY PAYMENT DUE
245Ø '
2455 GOSUB 69Ø:LOCATE 11,17
246Ø IF TR=1 THEN PRINT P$(L1);:ELSE PRINT P$(L2);
2465 COLOR 2:PRINT"... Your BILL is DUE!";:COLOR 1
247Ø UBILL=4*(INT(RND(1)*12)+1)
2475 IF TR=1 AND O(12)=2 AND O(28)=2 THEN UBILL=5*UBILL
248Ø IF TR=2 AND O(12)=1 AND O(28)=1 THEN UBILL=5*UBILL
2485 IF TR=1 THEN M(1)=M(1)-UBILL:M(2)=M(2)+UBILL
249Ø IF TR=2 THEN M(2)=M(2)-UBILL:M(1)=M(1)+UBILL
2495 LOCATE 12,17:PRINT"The Outrageous Charge IS......$";UBILL;
25ØØ PLAY"V15L64O1F":LOCATE 7,23:PRINT M(1);:PLAY"V15L64O1F":LOCATE 7,53:PRINT M
(2);
25Ø5 FOR DL=1 TO 9ØØ:NEXT:RETURN
251Ø '
2515 '*** PASS GO ***
252Ø '
2525 LOCATE 21,32:COLOR 3:PRINT"PASS GO! Get $2ØØ";:COLOR 1
253Ø IF TR=1 THEN M(1)=M(1)+2ØØ ELSE M(2)=M(2)+2ØØ
2535 IF TR=1 THEN LOCATE 7,23:PRINT M(1);:ELSE LOCATE 7,53:PRINT M(2);
254Ø PLAY"V15L64O5CO4CO3C":FOR DL=1 TO 8ØØ:NEXT:LOCATE 21,32:PRINT"
    ";:COLOR 1
2545 RETURN
255Ø '
2555 '*** PASS OUT ALL PROPERTY ***
256Ø '
2565 GOSUB 69Ø:LOCATE 11,17:COLOR 2:PRINT"SELECTING PROPERTIES";:COLOR 1:LOCATE
12,28,Ø:PRINT N$(1);:LOCATE 13,28,Ø:PRINT N$(2);
257Ø FOR PX=1 TO 14
2575 XX=INT(RND(1)*39)+1:IF O(XX)=Ø THEN O(XX)=1:LOCATE 12,36,Ø:PRINT P$(XX);:EL
SE 2575
258Ø PLAY"O3T64L64F"
2585 XY=INT(RND(1)*39)+1:IF O(XY)=Ø THEN O(XY)=2:LOCATE 13,36,Ø:PRINT P$(XY);:EL
SE 2585
259Ø PLAY"O5T64L64F"
2595 LOCATE 12,36:PRINT STRING$(25," ");:LOCATE 13,36:PRINT STRING$(25," ");
26ØØ NEXT PX
26Ø5 FOR X=1 TO 4Ø:IF O(X)=1 THEN 261Ø ELSE 2615
261Ø PRP(1)=PRP(1)+MV(X):IF (X=5 OR X=15 OR X=25 OR X=35) THEN RROWN(1)=RROWN(1)
+1
2615 NEXT
262Ø FOR X=1 TO 4Ø:IF O(X)=2 THEN 2625 ELSE 263Ø
2625 PRP(2)=PRP(2)+MV(X):IF (X=5 OR X=15 OR X=25 OR X=35) THEN RROWN(2)=RROWN(2)
+1
263Ø NEXT
```

```
2635 TR=1:GOSUB 86Ø
264Ø TR=2:GOSUB 86Ø
2645 GOTO 385
265Ø '
2655 *** MOVING THE PLAYING PIECES ***
266Ø '
2665 IF L1<> L2 THEN PUT(O1(L1),D1(L1)),U:ELSE FL=Ø
267Ø PUT(O1(L1),D1(L1)),U,PSET
2675 FOR X=1 TO (RL1+RL2)
268Ø L1=L1+1:IF L1> 4Ø THEN GOSUB 2525
2685 IF L1>=41 THEN L1=L1-4Ø
269Ø IF L1=L2 THEN FOR DL=1 TO 1ØØ:NEXT:PLAY"V1ØL64O2CO5C":FOR DL=1 TO 75:NEXT:G
OTO 272Ø
2695 GET(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),U
27ØØ LINE(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),2,BF
27Ø5 LINE(O1(L1)+1,D1(L1)+1)-(O1(L1)+14,D1(L1)+7),1,B
271Ø PLAY"V12T64L64O2CO5C":FOR DL=1 TO 75:NEXT
2715 PUT(O1(L1),D1(L1)),U,PSET
272Ø NEXT X
2725 IF L1<>L2 THEN LINE(O1(L1),D1(L1))-(O1(L1)+15,D1(L1)+8),2,BF
273Ø IF L1<>L2 THEN LINE(O1(L1)+1,D1(L1)+1)-(O1(L1)+14,D1(L1)+7),1,B
2735 RETURN
274Ø IF L2<>L1 THEN PUT(O1(L2),D1(L2)),W:ELSE FK=Ø
2745 PUT(O1(L2),D1(L2)),W,PSET
275Ø FOR X=1 TO (RL1+RL2)
2755 L2=L2+1:IF L2> 4Ø THEN GOSUB 2525
276Ø IF L2>=41 THEN L2=L2-4Ø
2765 IF L2=L1 THEN FOR DL=1 TO 1ØØ:NEXT:PLAY"V1ØL64O2CO5C":FOR DL=1 TO 75:NEXT:G
OTO 2795
277Ø GET(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),W
2775 LINE(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),3,BF
278Ø LINE(O1(L2)+1,D1(L2)+1)-(O1(L2)+14,D1(L2)+7),1,B
2785 PLAY"V12T64L64O2CO5C":FOR DL=1 TO 75:NEXT
279Ø PUT(O1(L2),D1(L2)),W,PSET
2795 NEXT X
28ØØ IF L2<>L1 THEN LINE(O1(L2),D1(L2))-(O1(L2)+15,D1(L2)+8),3,BF
28Ø5 IF L2<>L1 THEN LINE(O1(L2)+1,D1(L2)+1)-(O1(L2)+14,D1(L2)+7),1,B
281Ø RETURN
2815 '
282Ø '*** ROLLED A DOUBLE ***
2825 '
283Ø LOCATE 17,33:PRINT" DOUBLE ROLLED! ";:FOR DL=1 TO 15ØØ:NEXT:PLAY"V15L64O3D"
:LOCATE 17,33:PRINT STRING$(16," ");
2835 IF TR=1 THEN TR=TR+1 ELSE TR=1
284Ø DB=Ø:RETURN
2845 '
285Ø '*** SUMMARY WINDOW ***
2855 '
286Ø GET(5Ø,35)-(2ØØ,15Ø),V
2865 LINE(5Ø,35)-(2ØØ,15Ø),Ø,BF
287Ø LINE(51,36)-(199,149),3,B
2875 LOCATE 6,8:COLOR 2:PRINT"REAL ESTATE:";
288Ø EN=Ø:XX=8:COLOR 1
2885 FOR PROP=1 TO 4Ø:LOCATE XX,8:PLAY"V15L64O2F"
289Ø IF O(PROP)=Ø THEN PRINT LEFT$(P$(PROP),8);:COLOR 1:PRINT "-BNK";:COLOR 1:EN
=EN+1:XX=XX+1:COLOR 1
2895 IF O(PROP)=1 THEN PRINT LEFT$(P$(PROP),8);:COLOR 2:PRINT"-";LEFT$(N$(1),3);
:COLOR 1:EN=EN+1:XX=XX+1:IF B(PROP)=1 THEN COLOR 3:PRINT"-Book";:COLOR 1
2896 IF O(PROP)=3 THEN PRINT LEFT$(P$(PROP),8);:COLOR 2:PRINT"-";LEFT$(N$(1),3);
```

```
:COLOR 1:EN=EN+1:XX=XX+1:COLOR 3:PRINT"=Mrtg";:COLOR 1
29ØØ IF O(PROP)=2 THEN PRINT LEFT$(P$(PROP),8);:COLOR 3:PRINT"-";LEFT$(N$(2),3);
:COLOR 1:EN=EN+1:XX=XX+1:IF B(PROP)=2 THEN COLOR 2:PRINT"=Book";:COLOR 1
29Ø1 IF O(PROP)=4 THEN PRINT LEFT$(P$(PROP),8);:COLOR 3:PRINT"-";LEFT$(N$(2),3);
:COLOR 1:EN=EN+1:XX=XX+1:COLOR 2:PRINT"=Mrtg";:COLOR 1
29Ø5 IF EN=1Ø THEN EN=EN-1Ø:XX=XX-1Ø:GOTO 2915
291Ø GOTO 294Ø
2915 LOCATE 18,8:PRINT"< Any Key >";
292Ø AK$=INKEY$:IF AK$=""THEN 292Ø
2925 LINE(5Ø,35)-(2ØØ,15Ø),Ø,BF
293Ø LINE(51,36)-(199,149),3,B
2935 LOCATE 6,8:PRINT"REAL ESTATE";
294Ø NEXT
2945 LOCATE 18,8:PRINT"Any KEY";:
295Ø AK$=INKEY$:IF AK$=""THEN 2945
2955 COLOR 1
296Ø PUT(5Ø,35),V,PSET
2965 GOTO 72Ø
297Ø '
2975 '** READ IN THE PROPERTY HERE **
298Ø '
2985 FOR X=1 TO 4Ø:READ P$(X),O(X),RØ(X),R1(X),R2(X),R3(X),R4(X),R5(X),HC(X),MV(
X),UN(X),P(X):NEXT X
299Ø DATA BALTIMORE AVE,Ø,2,1Ø,3Ø,9Ø,16Ø,25Ø,5Ø,3Ø,33,6Ø
2995 DATA COUNTY CHEST,5,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
3ØØØ DATA MEDIOCRE AVE,Ø,4,2Ø,6Ø,18Ø,32Ø,45Ø,5Ø,3Ø,33,6Ø
3ØØ5 DATA INCOME TAX,5,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
3Ø1Ø DATA READING RR,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,1ØØ,11Ø,2ØØ
3Ø15 DATA ORIENTAL AVE,Ø,6,3Ø,9Ø,27Ø,4ØØ,55Ø,5Ø,5Ø,55,1ØØ
3Ø2Ø DATA CHANCE,5,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
3Ø25 DATA VERMIN AVE,Ø,6,3Ø,9Ø,27Ø,4ØØ,55Ø,5Ø,5Ø,55,1ØØ
3Ø3Ø DATA CONNECT AVE,Ø,8,4Ø,1ØØ,3ØØ,45Ø,6ØØ,5Ø,6Ø,66,12Ø
3Ø35 DATA JAIL,5,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
3Ø4Ø DATA ST CHUCKS PL,Ø,1Ø,5Ø,15Ø,45Ø,625,75Ø,1ØØ,7Ø,77,14Ø
3Ø45 DATA ELECTRIC CO,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,75,82,15Ø
3Ø5Ø DATA STATES AVE,Ø,1Ø,5Ø,15Ø,45Ø,625,75Ø,1ØØ,7Ø,77,14Ø
3Ø55 DATA VIRGINIA AVE,Ø,12,6Ø,18Ø,5ØØ,7ØØ,9ØØ,1ØØ,8Ø,88,16Ø
3Ø6Ø DATA PENN RR,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,1ØØ,11Ø,2ØØ
3Ø65 DATA ST JAMES PL,Ø,14,7Ø,2ØØ,55Ø,75Ø,95Ø,1ØØ,9Ø,99,18Ø
3Ø7Ø DATA COUNTY CHEST,5,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
3Ø75 DATA TENNESSEE AVE,Ø,14,7Ø,2ØØ,55Ø,75Ø,95Ø,1ØØ,9Ø,99,18Ø
3Ø8Ø DATA NEW YORK AVE,Ø,16,8Ø,22Ø,6ØØ,8ØØ,1ØØØ,1ØØ,1ØØ,11Ø,2ØØ
3Ø85 DATA FREE PARKING,5,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
3Ø9Ø DATA KENTUCKY AVE,Ø,18,9Ø,25Ø,7ØØ,875,1Ø5Ø,15Ø,11Ø,121,22Ø
3Ø95 DATA CHANCE,5,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
31ØØ DATA INDIANA AVE,Ø,18,9Ø,25Ø,7ØØ,875,1Ø5Ø,15Ø,11Ø,121,22Ø
31Ø5 DATA ILLINOIS AVE,Ø,2Ø,1ØØ,3ØØ,75Ø,925,11ØØ,15Ø,12Ø,132,24Ø
311Ø DATA B & O RR,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,1ØØ,1ØØ,2ØØ
3115 DATA ATLANTIS AVE,Ø,22,11Ø,33Ø,8ØØ,975,115Ø,15Ø,13Ø,143,26Ø
312Ø DATA VENTNOR AVE,Ø,22,11Ø,33Ø,8ØØ,975,115Ø,15Ø,13Ø,143,26Ø
3125 DATA WATER WORKS,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,75,82,15Ø
313Ø DATA MARVIN GDNS,Ø,24,12Ø,36Ø,85Ø,1Ø25,12ØØ,15Ø,14Ø,154,28Ø
3135 DATA GO TO JAIL,5,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
314Ø DATA PACIFIC AVE,Ø,26,13Ø,39Ø,9ØØ,11ØØ,1275,2ØØ,15Ø,165,3ØØ
3145 DATA N CAROLINA AVE,Ø,26,13Ø,39Ø,9ØØ,11ØØ,1275,2ØØ,15Ø,165,3ØØ
315Ø DATA COUNTY CHEST,5,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
3155 DATA PENN AVE,Ø,28,15Ø,45Ø,1ØØØ,12ØØ,14ØØ,2ØØ,16Ø,176,32Ø
316Ø DATA SHIRT LINE RR,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,1ØØ,11Ø,2ØØ
3165 DATA CHANCE,5,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
```

```
3170 DATA PORK PLACE,Ø,35,175,5ØØ,11ØØ,13ØØ,15ØØ,2ØØ,175,192,35Ø
3175 DATA LUXURY TAX,5,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
318Ø DATA BIRDWALK,Ø,5Ø,2ØØ,6ØØ,14ØØ,17ØØ,2ØØØ,2ØØ,2ØØ,22Ø,4ØØ
3185 DATA GO,5,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø
319Ø RETURN
3195 '
32ØØ '** DRAW THE PLAYING BOARD **
32Ø5 '
321Ø ' HOW TO CREATE FOLLOWING LINES......The CHARACTERS are created by holding
3215 ' down the ALT key and typing in the corresponding CHR$ key on the Numeric
322Ø ' Keypad. For example to get a 'è' you HOLD ALT & type 2Ø1,  then release.
3225 '
323Ø ' ¿ = 185   ¤ = 186   £ = 187   ¥=188   à = 2ØØ   è = 2Ø1   ò = 2Ø2
3235 ' ù = 2Ø3   ä = 2Ø4   ë = 2Ø5   ö = 2Ø6
324Ø '
3245 '*** Don't PANIC...Your printer MAY REFUSE TO LIST THESE LINES.
325Ø '
3255 B1$="     èëëëëëùëëëëëùëëëëëùëëëëëùëëëëëùëëëëëùëëëëëùëëëëëùëëëëëùëëëëëùëëë
ëëë£"
326Ø B2$="     ú      ú      ú      ú      ú      ú      ú      ú      ú      ú      ú
   ú"
3265 B3$="     ú       äëëëëëöëëëëëöëëëëëöëëëëëöëëëëëöëëëëëöëëëëëöëëëëëöëëëëë¿
   ú"
327Ø B4$="     äëëëùëöëëëëëòëëëëëòëëëëëòëëëëëòëëëëëòëëëëëòëëëëëòëëëëëòëëëëëöëùë
ëëë¿"
3275 B5$="     ú    ú ú                                              ú ú
   ú"
328Ø B6$="     äëëëëöë¿                                              äëöë
ëëë¿"
3285 B7$="     äëëëòëöëëëëëùëëëëëùëëëëëùëëëëëùëëëëëùëëëëëùëëëëëùëëëëëùëëëëëöëòë
ëëë¿"
329Ø B8$="     àëëëëëëòëëëëëòëëëëëòëëëëëòëëëëëòëëëëëòëëëëëòëëëëëòëëëëëòëëëëëòëëë
ëëë¥"
3295 LOCATE 25,1:PRINT B8$;
33ØØ LOCATE 1,1,Ø:PRINT B1$:PRINT B2$:PRINT B3$:PRINT B4$:PRINT B5$:PRINT B6$:PR
INT B5$:PRINT B6$:PRINT B5$:PRINT B6$:PRINT B5$:PRINT B6$:PRINT B5$:PRINT B6$:PR
INT B5$:PRINT B6$:PRINT B5$:PRINT B6$
33Ø5 PRINT B5$:PRINT B6$:PRINT B5$:PRINT B7$:PRINT B3$:PRINT B2$;
331Ø '
3315 '*** GETTING the Graphic Dice! ***
332Ø '
3325 LINE(258,14Ø)-(288,155),1,B:LINE(377,14Ø)-(347,155),1,B
333Ø PSET(272,147):PSET(357,145):PSET(367,15Ø):GET(258,14Ø)-(288,155),NU:GET(377
,14Ø)-(347,155),NV:PSET(263,143):PSET(283,151):GET(258,14Ø)-(288,155),NW:PSET(35
7,15Ø):PSET(367,145):GET(377,14Ø)-(347,155),NX
3335 PSET(362,147):GET(377,14Ø)-(347,155),NY:PRESET(272,147):PRESET(263,143):PRE
SET(283,151):PSET(268,144):PSET(268,148):PSET(268,152):PSET(278,144):PSET(278,14
8):PSET(278,152):GET(258,14Ø)-(288,155),NZ
334Ø '
3345 '*** Painting the Property ***
335Ø '
3355 PAINT(14Ø,175),"L":PAINT(2Ø5,175),"L":PAINT(31Ø,175),2,1:PAINT(345,175),2,1
:PAINT(441,175),2,1:PAINT(498,175),2,1
336Ø PAINT(14Ø,25),3,1:PAINT(2Ø5,25),3,1:PAINT(288,25),3,1:PAINT(31Ø,175),1,1:PA
INT(345,25),2,1:PAINT(441,25),2,1:PAINT(498,25),2,1:PAINT(31Ø,25),1,1
3365 PAINT(9Ø,163),3,1:PAINT(9Ø,137),3,1:PAINT(9Ø,98),1,1:PAINT(9Ø,8Ø),"z":PAINT
(9Ø,55),"z":PAINT(9Ø,3Ø),"z"
337Ø PAINT(545,163),3,1:PAINT(545,137),3,1:PAINT(545,11Ø),3,1:PAINT(545,98),1,1:
PAINT(545,8Ø),"M":PAINT(545,7Ø),"M":PAINT(545,3Ø),"M"
```

```
3375 LINE(115,125)-(24Ø,16Ø),1,B:LINE(515,125)-(39Ø,16Ø),1,B:LINE(115,35)-(515,7
Ø),1,B:LINE(115,72)-(515,1Ø7),1,B:PAINT(113,125),2,1
338Ø LINE(12Ø,128)-(235,157),1,B:LINE(51Ø,128)-(395,157),1,B:PAINT(117,126),3,1:
PAINT(512,126),3,1
3385 LINE(12Ø,75)-(51Ø,1Ø4),1,B:LINE(12Ø,38)-(51Ø,67),1,B:PAINT(119,76),3,1:PAIN
T(119,4Ø),3,1
339Ø LOCATE 23,8:PRINT CHR$(24);"GO":LOCATE 23,68:PRINT"GO TO":LOCATE 24,68:PRIN
T"JAIL!";:LOCATE 24,8:PRINT CHR$(192);CHR$(196);CHR$(196);
3395 LOCATE 24,14:PRINT"BWALK";:LOCATE 24,21:PRINT "TAX";:LOCATE 24,26:PRINT"PLA
CE";:LOCATE 24,34:PRINT"?";:LOCATE 24,38:PRINT"SL RR";:LOCATE 24,44:PRINT"PA AV"
;:LOCATE 24,52:PRINT"$";:LOCATE 24,56
34ØØ PRINT"NC AV";:LOCATE 24,62:PRINT"PC AV";
34Ø5 LOCATE 2,8:PRINT"JAIL":LOCATE 3,8:PRINT"BIRD":LOCATE 2,69:PRINT"FREE":LOCAT
E 3,69:PRINT"PARK"
341Ø LOCATE 2,14:PRINT"ST CH":LOCATE 2,2Ø:PRINT"ELECT":LOCATE 2,26:PRINT"STATE":
LOCATE 2,32:PRINT"VA AV":LOCATE 2,38:PRINT"-PRR-":LOCATE 2,44:PRINT"ST JA":LOCAT
E 2,51:PRINT" $ ":LOCATE 2,56
3415 PRINT"TENN.":LOCATE 2,62:PRINT"NY AV"
342Ø LOCATE 5,7:PRINT"CONN":LOCATE 7,7:PRINT"VERM":LOCATE 9,7:PRINT" ?":LOCATE 1
1,7:PRINT"ORNT":LOCATE 13,7:PRINT"RRR-":LOCATE 15,7:PRINT"$2ØØ"
3425 LOCATE 17,7:PRINT"MEDI":LOCATE 21,7:PRINT"BALT":LOCATE 19,7:PRINT" $"
343Ø LOCATE 5,7Ø:PRINT"KENT":LOCATE 9,7Ø:PRINT"IND.":LOCATE 11,7Ø:PRINT"ILL.":LO
CATE 7,7Ø:PRINT" ?":LOCATE 13,7Ø:PRINT"B&O"
3435 LOCATE 15,7Ø:PRINT"ATLA":LOCATE 17,7Ø:PRINT"VENT":LOCATE 21,7Ø:PRINT"MARV":
LOCATE 19,7Ø:PRINT"WATR":LOCATE 8,24:PRINT"WAIT-I'm passing out the money!"
344Ø PAINT(5Ø,19Ø),2,1:PAINT(5Ø,15),3,1:PAINT(585,19Ø),3,1:PAINT(585,15),2,1:PAI
NT(62Ø,1Ø),CHR$(53)
3445 LOCATE 17,32:PRINT STRING$(17," "):LOCATE 21,32:PRINT STRING$(17," "):LOCAT
E 18,52:PRINT" C H E S T":LOCATE 18,17:PRINT"C H A N C E"
345Ø LOCATE 15,31:PRINT" T A N Y O P O L Y "
3455 TT$="Tanyopoly by L Hyre"
346Ø FOR TT=4 TO 22:LOCATE TT,2,Ø:PRINT " "+MID$(TT$,TT-3,1)+" ";:NEXT:LOCATE 8,
24:PRINT STRING$(34," ")
3465 RETURN
347Ø '
3475 '** READ IN COUNTY CHEST AND CHANCE HERE **
348Ø '
3485 FOR CHANCE=1 TO 12:READ C1$(CHANCE),C2$(CHANCE):NEXT
349Ø DATA JAYWALKER,$25 FINE!,$25 FOR,EACH HSE/HTL,GET OUT OF,JAIL FREE!,ADVANCE
 TO,NEAREST RR,PAY!  EACH,PLAYER $5Ø,ADVANCE TO,ST CHUCKS,ADVANCE TO,--- GO ---,
ADVANCE TO--,ILLINOIS AVE,GO TO JAIL,*DIRECTLY*
3495 DATA ADVANCE TO-,*BIRDWALK*,LOAN MATURES,COLLECT- $5Ø,PAY POOR TAX,--- $ 15
 ---
35ØØ FOR CHEST=1 TO 12:READ C4$(CHEST),C5$(CHEST):NEXT
35Ø5 DATA COLLECT $5Ø,FROM FRIEND,COLLECT $45,FROM  STOCK,ADVANCE TO,--- GO ---,
BANK ERROR!,COLLECT $2ØØ,RECEIVE $25,FOR SERVICES,GO TO JAIL,-DIRECTLY!-,PAY $1Ø
Ø ,HOSPITAL,SOLD C-64,COLLECT $2Ø,SCHOOL TAX,PAY   $15Ø,XMAS  CLUB,COLLECT $1ØØ
351Ø DATA DOCTOR FEE,** $ 5Ø **,CONTEST,WIN $12
3515 RETURN
352Ø RL=INT(RND(1)*6)+1:PRINT RL:FOR DL=1 TO 2ØØ:NEXT:GOTO 352Ø
3525 '
353Ø '*** WHERE TO PUT PLAYING PIECES ***
3535 '
354Ø FOR LP=1 TO 4Ø:READ O1(LP),D1(LP):NEXT
3545 DATA 53,158,53,141,53,126,53,11Ø,53,93,53,78,53,62,53,46,53,3Ø,53,8,113,6,1
61,6,2Ø9,6,257,6,3Ø5,6,352,6,4Ø2,6,451,6,497,6,545,6,545,3Ø,545,46,545,62,545,78
,545,93,545,11Ø,545,126,545,141,545,158,545,18Ø,496,18Ø,451,18Ø,4Ø1,18Ø,353,18Ø
355Ø DATA 3Ø7,18Ø,257,18Ø,2Ø9,18Ø,158,18Ø,113,18Ø,57,18Ø
3555 RETURN
```

PCM

Putting the squeeze on data files

Squeezing Bytes on Disk

By William Barden, Jr.
PCM Contributing Editor

An old programmer's axiom (and I'm one of the oldest programmers you know) goes something like this: "The amount of available memory is always 1K less than what is required." A corollary says just about the same thing for disk storage. RAM is getting less and less expensive — about a dollar per kilobyte. Hard disk prices are also falling — about 4.5 cents per kilobyte.

In spite of these trends, we're going to look at some software alternatives to packing more data onto disk files in this month's column. What tricks can be used to pack more data onto disk? What are some of the data compression methods? What in the world is Huffman encoding? Regardless of the amount of main memory or disk storage, you'll find many cases where the memory axiom and corollary raise their ugly heads.

William Barden, Jr. is a master communicator in a field in which he is one of the few recognized experts — microcomputers. A prolific author of more than 27 books on computers and computer programming, Bill also has authored several instructional software projects for Tandy/Radio Shack.

Sequential Files

The place to begin is with an old friend, the sequential file. Several months ago I did a column on sequential files. In it, I told you I liked them because they are "user friendly." With sequential files, you generally know what you have — you can do a TYPE command in MS-DOS and see the entire file printed without strange happy faces followed by control codes that cause screen clears and other unpredictable actions.

Although it's possible to incorporate control codes (those characters with a value of less than 32 decimal)

or the extended character set (those characters with a value greater than 127 decimal), sequential files are usually ASCII files made up of alphabetic, numeric and special characters between 32 (space) and 127 (backspace).

Take the following MS-DOS sequence, for example:

```
A>copy con scratch
They can have my Tandy 1000 when they pry my cold, dead
fingers from it!
Tippicanoe and Tandy too!
^Z
```

This sequence builds an ASCII (sequential) file called SCRATCH made up of three lines. (I had to wrap the first line around in the text above, but not on the screen.) The file can be examined using *Debug*:

```
A>debug scratch
-d
0931:0100 54 68 65 79 20 ....65 20 6D 79 They can have my
.
.
.
```

The first column represents the memory location into which the file has been loaded (it may be different on your system). The bulk of the display consists of hexadecimal data showing the characters in sequence in the file. The right-hand portion of the display shows the corresponding ASCII characters, or periods if there are no displayable ASCII characters for the data.

Between the first and second lines of the *Debug* dump, you'll see:

```
0931:0140 66 72 6F 6D 20 69 74 21-0D 0A 74 69 70 70 69 63
```

for the location and Hex data. The 0D 0A represents the carriage return and line feed that make up the end of the first line.

Processing Sequential Files with BASIC

The nice thing about sequential files is that you can read in the files in BASIC and process them anyway you'd like. For example, this code would read in and print each line of the file plus the line length:

```
100 OPEN "SCRATCH" FOR INPUT AS #1
110 WHILE NOT EOF( 1 )
120     LINE INPUT#1, A$
130     PRINT A$; "/"; LEN( A$ )
140 WEND
```

resulting in

```
They can have my Tandy 1000 when they pry my cold, dead
fingers from it!/ 72
Tippicanoe and Tandy too!/ 25
```

Creating Sequential Files with BASIC

BASIC can process any ASCII file created by an MS-DOS COPY command, by a word processor, by telecommunications or other means. BASIC, of course, can also *create* sequential files. In a BASIC sequential file, *all* data, even numeric data, is made up of ASCII characters.

Suppose you want to create an inventory file with part number, description and quantity on hand, denoted by variables PN, DESC$ and QOH. Here's one way to do it, working with keyboard entry data:

```
100 OPEN "INVENT" FOR OUTPUT AS #1
110 INPUT PN, DESC$, QOH
120 PRINT#1, PN, DESC$, QOH
```

A typical file created by this method is shown in Figure 1.

Notice the three parts take up 134 bytes of disk storage, counting carriage return/line feeds (0D/0A) and end-of-file marker (1A), but only 81 characters are represented:

```
100 Write-Only Memories 1234
101 No-Sided Disk 125
102 Archer Nuclear Device Kit 75
```

This is a 40 percent waste of storage in this short file!

A better way to store this data is by using PRINT# with semicolons. See Figure 2.

Here, 94 bytes were used for 82 bytes of actual data and only 13 percent of the file space was wasted, for carriage return/line feeds (0D/0A), spaces for padding (before and after numeric values) and the 1A end-of-file character.

By the way, in this discussion we're ignoring the fact that BASIC and MS-DOS allocate space for disk files in blocks of 1,024 bytes. Obviously, if files are very short, there's going to be wasted space in the unused portion of the file. For typical large files, however, the unused space at the end of the last 1,024 bytes becomes less and less of a factor. A file of 65,000 characters, for example, uses 64 1,024-byte "clusters." The 64th cluster uses only 488 bytes, but the overall wasted space is only 536/65,536 or 0.8 percent. We're talking about large files here, but using short segments of data for examples.

A third way of storing data in a sequential file is by using WRITE# statements. WRITE# is specifically geared to using minimum space on files. See Figure 3 for the modified code and the resultant file.

Here, 94 bytes were also used, the reduction in padding spaces being offset by automatic insertion of commas and double quotes.

There Must Be a Better Way

Is that the best we can do? There must be a more efficient way to store data.

Right at the outset, you can guess that there is, but *compressing* data on disk will not come cheaply, either in time or user friendliness.

Random Files

Random files are an alternative to sequential files. They should be used any time a file is not read through from beginning to end — the inventory example might access an unpredictable sequence of part numbers, for example — Part Number 34 followed by Part Number 105, followed by Part Number 2, etc.

Random files, though, don't usually save disk space over sequential files. As each *record* in a random file is fixed length, chances are much space is wasted, as the largest possible record size must be accounted for. In the sequence

```
100 Write-Only Memories 1234
101 No-Sided Disk 125
102 Archer Nuclear Device Kit 75
```

for example, the record size would probably be large enough to handle the last entry, as it's the longest.

On the other hand, random files with little text data and a lot of numeric data could conceivably result in some data

compression, as numeric values are stored in the record fields as numeric and not ASCII data. The value 104.56, for example, would be stored as a single-precision four-byte number instead of the six characters 1, 0, 4, ., 5 and 6. Random files are not the answer.

An Old Programmer's Trick

One method of data compression uses the following scheme: two bytes can hold values of zero to 65,535, unsigned. (BASIC integer values can be –32,768 through +32,767 in signed form, also in two bytes.) Suppose we use the two bytes to represent *three* digits with the *base* of 40. We'll use the following codes:

A	0	U	20
B	1	V	21
C	2	W	22
D	3	X	23
E	4	Y	24
F	5	Z	25
G	6	0	26
H	7	1	27
I	8	2	28
J	9	3	29
K	10	4	30
L	11	5	31
M	12	6	32
N	13	7	33
O	14	8	34
P	15	9	35
Q	16	space	36
R	17	.	37
S	18	new line	38
T	19	special	39

As long as data can be represented by these 40 characters, this "base 40 packing" scheme enables you to boldly pack three characters where two have gone before. Suppose, for example, you wanted to store the word "THE." In this coding scheme, the representation would be:

$$T * 40^2 + H * 40^1 + E * 40^0 =$$
$$19 * 40^2 + 7 * 40^1 + 4 * 40^0 =$$
$$30400 + 280 + 4 = 30,684$$

The characters could be "unpacked" by a "divide and save remainders" technique:

30,684/40 = 767, remainder 4
767/40 = 19, remainder 7
19/40 = 0, remainder 19

To see how this scheme works, look at listings 1 and 2. They show a packing routine that converts an ASCII file to a "base-40-packed" file and an unpacking routine that reconverts the packed file to an ASCII file.

TOLL FREE ORDER LINE
(800) 245-6228

SPECIAL !!!

SALE ON HARD DRIVE SYSTEMS FOR THE TANDY 1000

INCLUDES WD CONTROLLER CABLES, MOUNTING HARDWARE & INSTRUCTIONS

10 MEG....$385.

20 MEG....$489.

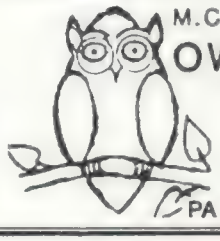
EXTERNAL CASE AND POWER SUPPLY ADD...$125.

20 MEG 1/2 LENGTH HARD CARD $655.
This Allows TWO FLOPPIES Internally!!!

Memory Card with 512K DMA $149.
Multifunction Card $99.

1 Year Warranty on ALL Items!!!

Call for LATEST PRICES!!! (800) 245-6228



M.C. & VISA Accepted
OWL-Services
P.O. Box 116-D
Mertztown, PA.
19539
PA Res Include 6% Tax
PA (215) 682-6855

The blood of Kings or of Knaves or both. Family Roots software helps you trace your bloodlines back through lost generations and the mists of time.

For your Apple, IBM, Commodore, Kaypro, TRS-80,* and many others, Family Roots offers all kinds of genealogical assistance: Individual and group sheets, charts, name indices, general search and text capabilities. Adapts to most disk drives, printers, and screens. You get more utility programs, plus lots of personal control. A comprehensive (new) manual is included.

Blood.

All for just $185.

Write or call today for more information and a free brochure.

Quinsept, Inc.
P.O. Box 216
Lexington, MA 02173
(617) 641-2930

American Express, Visa, and MasterCard gladly accepted.

Family Roots

* *Trademarks for Apple Computer Inc., International Business Machines, CBM, Inc., Digital Research and Tandy Corp.*

Figure 1

```
A>debug invent
-d
0931:0100  20 31 30 30 20 20 20 20-20 20 20 20 20 20 57 72    100          Wr
0931:0110  69 74 65 2D 4F 6E 6C 79-20 4D 65 6D 6F 72 69 65    ite-Only Memorie
0931:0120  73 20 20 20 20 20 20 20-20 20 20 31 32 33 34 20    s          1234
0931:0130  0D 0A 20 31 30 31 20 20-20 20 20 20 20 20 20 20    .. 101
0931:0140  4E 6F 2D 53 69 64 65 64-20 44 69 73 6B 20 20 31    No-Sided Disk  1
0931:0150  32 35 20 0D 0A 20 31 30-32 20 20 20 20 20 20 20    25 .. 102
0931:0160  20 20 20 41 72 63 68 65-72 20 4E 75 63 6C 65 61       Archer Nuclea
0931:0170  72 20 44 65 76 69 63 65-20 4B 69 74 20 20 20 20    r Device Kit
-d
0931:0180  37 35 20 0D 0A 1A 3A C6-74 0C 3A C2 74 08 E8 8B    75 ...:Ft.:Bt.h.
```

Figure 2

```
100 OPEN "INVENT" FOR OUTPUT AS #1
110 INPUT PN, DESC$, QOH
120 PRINT#1, PN; DESC$; QOH

A>debug invent
-d
0931:0100  20 31 30 30 20 57 72 69-74 65 2D 4F 6E 6C 79 20     100 Write-Only
0931:0110  4D 65 6D 6F 72 69 65 73-20 31 32 33 34 20 0D 0A    Memories 1234 ..
0931:0120  20 31 30 31 20 4E 6F 2D-53 69 64 65 64 20 44 69     101 No-Sided Di
0931:0130  73 6B 20 31 32 35 20 0D-0A 20 31 30 32 20 41 72    sk 125 .. 102 Ar
0931:0140  63 68 65 72 20 4E 75 63-6C 65 61 72 20 44 65 76    cher Nuclear Dev
0931:0150  69 63 65 20 4B 69 74 20-37 35 20 0D 0A 1A 8A C2    ice Kit 75 ....B
0931:0160  3C 2C 8A C1 74 03 E8 B3-00 E8 8F 71 72 28 3C 0A    <,.At.h3.h.qr(<.
0931:0170  74 EA 3C 0D 75 0C 8A C2-3C 20 74 15 3C 2C B0 0D    tj<.u..B< t.<,0.
```

Listing 1:

```
100 ' Convert ASCII Disk File to Base 40 Coded Disk File
110 KEY OFF
120 B40$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ." + CHR$(0) + CHR$(0) +
      "abcdefghijklmnopqrstuvwxyz"
130 CLS
140 LOCATE 1,16
150 PRINT "Convert ASCII Disk File to Base 40 Coded Disk File"
160 LOCATE 3,5: INPUT "Source File: ", SF$
170 LOCATE 4,5: INPUT "Dest File: ", DF$
180 OPEN SF$ FOR INPUT AS #1
190 OPEN DF$ AS #2 LEN=128
200 FIELD#2, 128 AS BLOCK$
210 BSTRING$="": BSLEN = 0: BYTE = 0: DIGIT = 1600
220 WHILE NOT EOF( 1 )
230    LINE INPUT#1, IN$
240    PRINT IN$
250    FOR I = 1 TO LEN ( IN$ ): INCHR$ = MID$ (IN$, I, 1 )
260       V = INSTR( B40$, INCHR$ )
270       IF V = 0 THEN 300
280       IF V > 40 THEN V = V - 40
290       OUTCHR = V - 1: GOSUB 380
300    NEXT I
310    OUTCHR = 38: GOSUB 380
320 WEND
330 BYTE = BYTE + 39 * DIGIT: GOSUB 410
340 IF BSLEN <> 0 THEN LSET BLOCK$ = BSTRING$: PUT#2
350 CLOSE
360 GOTO 130
```

```
370 ' Subroutine to output the next Base 40 character
380 BYTE = BYTE + OUTCHR * DIGIT
390 IF DIGIT <> 1 THEN DIGIT = DIGIT / 40 ELSE GOSUB 410
400 RETURN
410 ' Subroutine to output next byte
420 DIGIT = 1600: BSTRING$ = BSTRING$ + CHR$( INT( BYTE / 256 ) ) +
    CHR$( BYTE - INT( BYTE / 256 ) * 256 ): BYTE = 0: BSLEN = BSLEN + 2
430 IF BSLEN = 128 THEN LSET BLOCK$ = BSTRING$: PUT#2: BSTRING$ = "":
    BSLEN = 0
440 RETURN
```

Listing 2:

```
100 ' Convert Base 40 Codes Disk File to ASCII Disk File
110 KEY OFF
120 B40$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 .*/"
130 CLS
140 LOCATE 1, 16
150 PRINT "Convert Base 40 Codes Disk File to ASCII Disk File"
160 'Open source and destination files
170 LOCATE 3, 5: INPUT "Source File: ", SF$
180 LOCATE 4, 5: INPUT "Dest File: ", DF$
190 OPEN SF$ AS #1 LEN=128
200 OPEN DF$ FOR OUTPUT AS #2
210 FIELD#1, 128 AS BLOCK$
220 'Read in and convert file
230 ASCII$ = "": BIN$ = "": TERM = 0
240 WHILE NOT TERM
250    GET#1: BSTRING$ = BLOCK$
260    FOR I = 1 TO LEN ( BSTRING$ ) STEP 2
270       BYTE1$ = MID$( BSTRING$, I, 1 )
280       BYTE2$ = MID$( BSTRING$, I + 1, 1 )
290       WORD = ASC( BYTE1$ ) * 256 + ASC( BYTE2$ )
300       CODE2 = INT( WORD / 1600 )
310       CODE1 = INT( ( WORD - CODE2 * 1600 ) / 40 )
320       CODE0 = WORD - ( CODE2 * 1600 ) - ( CODE1 * 40 )
330       CODE2$ = MID$( B40$, CODE2 + 1, 1 )
340       CODE1$ = MID$( B40$, CODE1 + 1, 1 )
350       CODE0$ = MID$( B40$, CODE0 + 1, 1 )
360       CCODE$ = CODE2$: GOSUB 440: IF TERM THEN 400
370       CCODE$ = CODE1$: GOSUB 440: IF TERM THEN 400
380       CCODE$ = CODE0$: GOSUB 440: IF TERM THEN 400
390    NEXT I
400 WEND
410 IF ASCII$ <> "" THEN PRINT#2, ASCII$: PRINT ASCII$: ASCII$ = ""
420 CLOSE
430 GOTO 130
440 ' Subroutine to store current character, test for new lines, and terminator
450 IF CCODE$ <> "*" THEN IF CCODE$ <> "/" THEN ASCII$ = ASCII$ + CCODE$:
    GOTO 480
460 IF CCODE$ = "*" THEN PRINT#2, ASCII$: PRINT ASCII$: ASCII$ = "":
    GOTO 480
470 TERM = -1: PRINT#2, ASCII$: PRINT ASCII$
480 RETURN
```

The packing program works like this: The ASCII (sequential) file must contain only alphabetic characters A-Z, a-z, digits 0-9, blanks and periods. A space is substituted for all other characters. The program reads in the ASCII file one line at a time. Each line is then scanned from left to right for characters. As each character is encountered, it is converted to a base-40 character from the B40$ array. Three of these characters at a time are packed into a single-byte value from zero to 63,999. The single-byte value is then stored into the next field of a *random* file. After each 128 bytes, a PUT is done to write out a random file record. A "new line" character (38) is packed at the end of each input file line. The process continues until the end of the input file, at which time the special character (39) is written out to mark the end of the file.

Why is a random file used? It would be possible to use a sequential file except for one glitch: BASIC sequential files use a terminating 1A (hexadecimal) character to mark the end of file. It's possible a 1A character may be generated in constructing the data, as in the sequence "MA1," which produces the two Hex bytes C0 and 1A. This erroneous end of file stops a read of the packed file in the unpacking program to be described. There's no way around the 1A end of file except to use a random file, which expects and allows binary data. (By specifying a single field and using every byte in the field without blanks for padding, we can

eliminate wasted space in the file.)

The first portion of a packed file is shown in Figure 4. It contains the three inventory entries as before, but they are not recognizable! That's one of the disadvantages of packing data in this fashion — you can't just simply scan a file using TYPE (which produces garbage) or *Debug*.

The unpacking program takes the packed random file and reads in (using GET) one 128-byte record at a time. As each record is read in, it is scanned as an ordinary string. Each byte of the string is reconverted to three base-40 digits, each digit having a value of zero through 39. The three digits are reconverted to the ASCII characters A-Z, 0-9, blank or period. A code of 38 results in a carriage return/line feed action. If the special character 39 is encountered it marks the end of file. This delimiter is necessary because the end of file may come anywhere within a random file record, unlike an ordinary random file record.

The pack and unpack programs work relatively fast — 11 seconds to process 1,000 characters when using a compiled version of the program. Data compression is 31 percent — each file is compressed to 67 percent of the best case sequential or random file, not considering the unused portion of the file due to MS-DOS file allocation. However, this is probably not worth the trouble. After all, we can't represent all of the 96 characters normally found in straight ASCII text.

Huffman Coding

In the base-40 coding and in ASCII encoding, each .pa character is assigned an equal "weight." Both the letter 'Q', an infrequently used letter, and the letter 'E', a frequently used letter, have an equal number of bits to represent them. It would make for a more efficient representation of data if characters were assigned bit lengths based upon their *frequency* of use. Morse code, for example, uses a single "dot" of one unit of time (an audible bleep) for the letter 'E' and the sequence "dash, dash, dot, dash" of 13 units of time for the infrequently used 'Q'. The result is a higher "Baud rate" for Morse code keying.

Huffman encoding uses such a frequency-dependent scheme. The letter 'E' may have a code length of four bits, while a 'Q' may have a code length of 10 bits.

To Huffman encode any set of characters, the following method is used. Start with a frequency table of characters. We'll use this table for ease of illustration:

a	142 times
h	40
j	1
m	20
n	57
q	0
v	11

Not all 26 alphabetic characters are here, of course, but the method works with any number of characters.

Now build a list of the characters in order, as shown in Figure 5. Find the two characters with the lowest frequency, in this example 'q' and 'j'. Connect the two characters with the branches of a "tree" and create a root with the new

CPYAT2PC™
Is Like Having A
FREE
360K Floppy Drive

NOT COPY PROTECTED

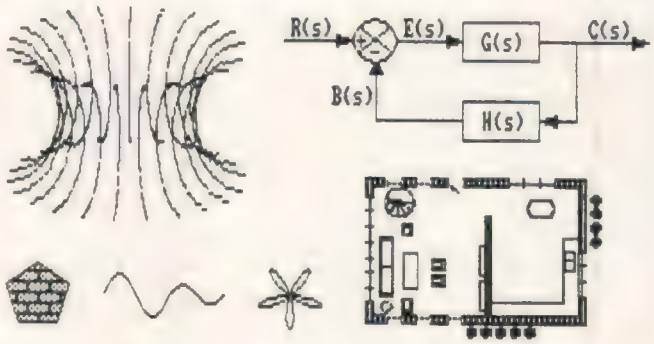
NOW AVAILABLE FOR THE TANDY 3000

Allows copying of TANDY 3000 hard disk files to the high density drive in 360K format. Files transferred may then be ***reliably*** read on PC's and Compatibles. A 360K floppy drive is ***not*** required. CPYAT2PC may reside on your TANDY 3000 hard disk and copies 1 file or entire subdirectories in 1 step. Also runs on other 80286 compatibles such as IBM PC/AT, COMPAQ 286, ZENITH Z-200 and HP VECTRA. Dealer inquiries welcome.
Only $79.00 + shipping.

PCM CERTIFICATION SEAL

MICROBRIDGE COMPUTERS
Sky Way Building, Suite 125
655 Sky Way, San Carlos, CA 94070
(415) 593-8777 • (415) 595-2150
CHECK, COD WELCOME
(Prepayment Required)
VISA® master charge

PC-KEY-DRAW
Version 3.0



High power graphics editor, CAD, paint, and slide show program for the Tandy 1000, 1200, and 3000 with 256K and color graphics card. Over 150 functions plus multiple pages multiple layers, macro feature, multiple text fonts, multiple patterns, mouse support, novice and expert modes, and more.

$15.00 trial price includes, complete program, partial manual on disk, and quick reference card.

$100.00 registration price includes, complete printed manual quick reference card, updates at cost and complete program.

Shareware: Copy & share. Send check or money order to:

oed
OEDWARE
P.O. Box 595
Columbia, MD 21045
(301) 997-9333

Figure 3

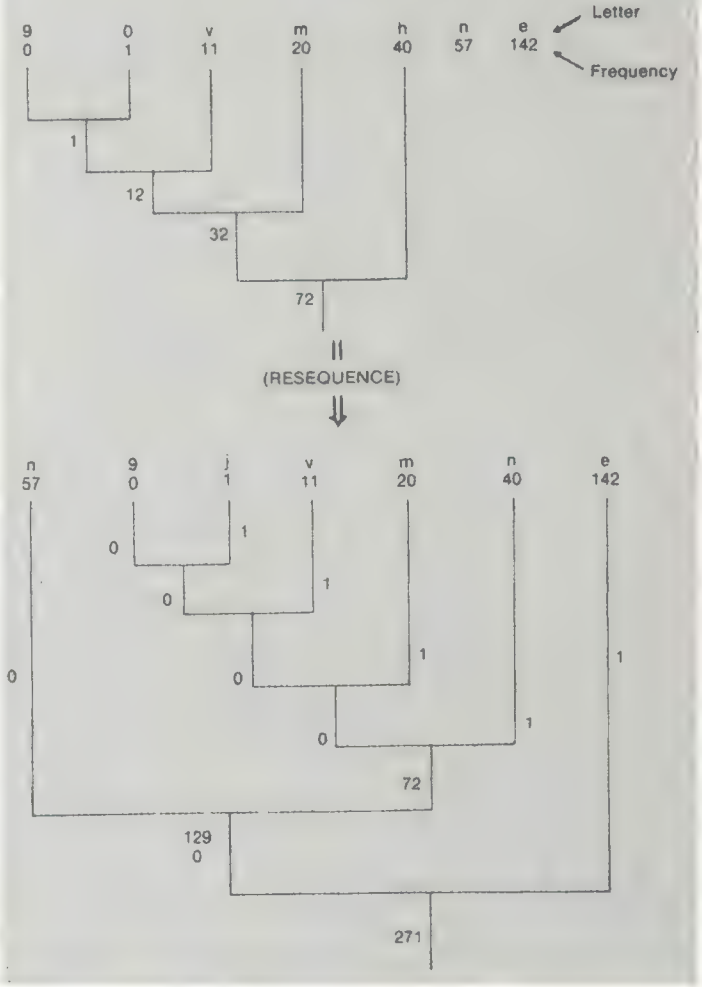
```
10Ø OPEN "INVENT" FOR OUTPUT AS #1
11Ø INPUT PN, DESC$, QOH
12Ø WRITE#1, PN, DESC$,QOH

A>debug invent
-d
Ø931:Ø1ØØ  31 3Ø 3Ø 2C 22 57 72 69-74 65 2D 4F 6E 6C 79 2Ø   1ØØ,"Write-Only
Ø931:Ø11Ø  4D 65 6D 6F 72 69 65 73-22 2C 31 32 33 34 ØD ØA   Memories",1234..
Ø931:Ø12Ø  31 3Ø 31 2C 22 4E 6F 2D-53 69 64 65 64 2Ø 44 69   1Ø1,"No-Sided Di
Ø931:Ø13Ø  73 6B 22 2C 31 32 35 ØD-ØA 31 3Ø 32 2C 22 41 72   sk",125..1Ø2,"Ar
Ø931:Ø14Ø  63 68 65 72 2Ø 4E 75 63-6C 65 61 72 2Ø 44 65 76   cher Nuclear Dev
Ø931:Ø15Ø  69 63 65 2Ø 4B 69 74 22-2C 37 35 ØD ØA 1A 8A C2   ice Kit",75.....B
Ø931:Ø16Ø  3C 2C 8A C1 74 Ø3 E8 B3-ØØ E8 8F 71 72 28 3C ØA   <,.At.h3.h.qr(<.
Ø931:Ø17Ø  74 EA 3C ØD 75 ØC 8A C2-3C 2Ø 74 15 3C 2C BØ ØD   tj<.u..B< t.<,Ø.
```

Figure 4

```
A>debug scratch
-d
ØCBF:Ø1ØØ  AC EA E4 81 34 FC 59 93-9B AC 1A EE 6B 84 76 3B   ,jd.4|Y..,.nk.v;
ØCBF:Ø11Ø  B3 A6 F1 D2 AE 6D 5A 58-13 63 E1 8Ø 72 34 AD 3F   3&qR.mZX.ca.r4-?
ØCBF:Ø12Ø  F1 D2 B4 AØ 6A 97 1B CC-54 62 45 6Ø 6F E3 1C 5Ø   qR4 j..LTbE'oc.P
ØCBF:Ø13Ø  ØD 44 3F D3 E6 47 F3 98-2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø   .D?SfGs.
ØCBF:Ø14Ø  2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø-2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
ØCBF:Ø15Ø  2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø-2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
ØCBF:Ø16Ø  2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø-2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
ØCBF:Ø17Ø  2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø-2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø 2Ø
```

Figure 5: Sample Huffman Code Tree



frequency, in this case, one. Find the two smallest frequencies again, considering even the new root, and creating another branch and root. In this case the "q j" root is combined with the 'v' root to create a root with a frequency of 12.

Continue the process until only one root remains, as shown in the figure. It helps to reorder the "subtrees" after each step, as shown. The resulting upside-down tree forms a series of *weighted* paths for each letter. The less often the letter appears, the longer the path will be and the greater the number of bits used to represent the letter.

For each branch to the left, assign a value of zero. For each branch to the right, assign a value of one. Now follow the main root back to each letter, building a bit sequence as you go. The result for this set of characters is shown below:

e	142 times	1
h	40	011
j	1	010001
m	20	0101
n	57	00
q	0	010000
v	11	01001

Note the 'e' is represented in only one bit, whereas the 'q' is represented in six bits. Also note each code is unique and no longer code contains a smaller code as the code bits are scanned from left to right.

The Huffman codes produced can now be put into a code table. This code table can then be used to convert ASCII characters to a series of bits. The text "even" would be represented as:

```
1/01001/1/00 = 101001100
```

This seems like a useful scheme of compressing data. In researching this column, I actually went through the process of producing a Huffman encoder and decoder routine. However, the end result was not too encouraging. The data compression for a 1,542-byte sequential file was 912 bytes, which is good — about a 41 percent compression. The code, however, was not trivial and moreover was rather slow, due to the bit shuffling in BASIC. Even the compiled version was barely acceptable as far as speed. If you would like to try this method of data compression, I strongly suggest using *Turbo* PASCAL, C or assembly language in place of BASIC.

A Frequency Dependent Algorithm

The Huffman approach shows promise. It appears the *repetition* of data should definitely be considered when compressing data. Let's investigate the frequency of characters in text files further.

The program shown in Listing 3 provides a count of characters versus their frequencies for any ASCII file. The results obtained for a typical PCM article are shown in Figure 6. It shows the bulk of characters are lowercase alphabetic characters — especially vowels. Blanks are the most common character. The 127 code here is used to represent the number of lines in the file (469).

Figure 6: Frequency Distribution of Typical ASCII File

```
.op
.lm 10
  :4458    0 : 339    @ :   0    P :  33    ` :   0    p : 226
! :   13   1 : 220    A : 123    Q :  11    a : 984    q :  60
" :   60   2 : 222    B :  52    R :  20    b : 197    r : 813
# :   12   3 : 162    C : 100    S :  78    c : 522    s : 818
$ :   10   4 : 111    D :  61    T : 100    d : 468    t :1046
% :    5   5 :  93    E :  53    U :  15    e :1699    u : 359
& :    0   6 : 127    F :  34    V :   5    f : 325    v :  94
' :   30   7 :  92    G :   4    W :  23    g : 210    w : 165
( :   25   8 :  39    H :  27    X :  81    h : 597    x :  35
) :   22   9 :  77    I : 105    Y :   3    i : 762    y : 200
* :    6   : :  47    J :   3    Z :   4    j :  10    z :   8
+ :    7   ; :   6    K :   7    [ :   0    k :  86    { :   0
, : 196    < :  10    L :   7    \ :   0    l : 496    | :   0
- :  73    = :   7    M :  19    ] :   0    m : 257    } :   0
. : 201    > :   5    N :  48    ^ :   8    n : 791    ~ :   0
/ :  12    ? :   6    O :  39    _ :   0    o : 856      : 469
```

Listing 3:

```
100 ' Build Frequency Table
110 KEY OFF
120 DIM FREQ ( 96 )
130 CLS
140 LOCATE 1, 16
150 PRINT "Build Frequency Table From Sample ASCII File"
160 LOCATE 3, 1: INPUT "File: ", SF$
170 PRINT
180 OPEN SF$ FOR INPUT AS #1
190 'Clear frequency table
200 FOR I = 32 TO 127
210    FREQ ( I - 32 ) = 0
220 NEXT I
230 'Read in a line at a time 'til end
240 WHILE NOT EOF( 1 )
250    LINE INPUT#1, INPTS$
260    ' Scan line, bump character count
270    FOR I = 1 TO LEN ( INPTS$ )
280       V = ASC( MID$( INPTS$, I, 1 ) )
290       IF V < 32 OR V > 127 THEN PRINT "Invalid character at "; I;
          "position of "; INPTS$: GOTO 310
300       FREQ( V - 32 ) = FREQ( V - 32 ) + 1
310    NEXT I
320    FREQ( 95 ) = FREQ( 95 ) + 1
330 WEND
340 CLOSE
350 'Display codes on screen
360 FOR I = 32 TO 47
370    PRINT USING "\\:####    "; CHR$(   I    ), FREQ( I - 32 );
380    PRINT USING "\\:####    "; CHR$( I + 16 ), FREQ( I - 16 );
390    PRINT USING "\\:####    "; CHR$( I + 32 ), FREQ(   I    );
400    PRINT USING "\\:####    "; CHR$( I + 48 ), FREQ( I + 16 );
410    PRINT USING "\\:####    "; CHR$( I + 64 ), FREQ( I + 32 );
420    PRINT USING "\\:####    "; CHR$( I + 80 ), FREQ( I + 48 )
430 NEXT I
500 END
```

The total number of characters here is 19,256. Amazingly enough, the 15 highest frequency characters amount to 14,686 characters, about 76 percent of the text in a typical ASCII file. This means we could encode two characters per byte (four bits per character) letting the values 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1010, 1011, 1100, 1101 and 1110 represent the 15 characters: space, 0, a, c, d, e, h, i, l, o, r, s, t, u and new line. (The code 1111 would be reserved for special purposes.)

This is a base-16 encoding scheme, analgous to the base-40 encoding seen earlier, but having the advantage that the "bit shuffling" to get at the codes is less complicated — just strip off the upper or lower four bits of every byte.

The special code 1111 could signify the character lies outside of these 15 most frequent characters. It would point to the next four bits, which would hold codes for the next 15 most frequent characters. From the frequency table, these characters represent about 14 percent of the characters found in a typical text file. Using this scheme, we could encode 76 out of 100 characters in about 38 bytes and 14 out of 100 characters in 14 bytes — 90 out of 100 characters in 52 bytes!

The remaining 10 percent of the characters, however, making up 66 various characters, would have to be encoded in an additional four bits or eight bits, adding about 17 bytes to the total for about 69 compressed characters out of 100 original characters. Still, this is about the same as the base-40 encoding used earlier, and we'd be able to represent all 96 ASCII characters (but without control codes).

Word Repetition

There are other repetitions to be considered as well, however. In text files, one obvious repetition is with words. The word "the" appears in this article about 280 times. "The" with a leading and trailing space takes up five bytes for every occurrence. Is there some way to encode this into a single byte or perhaps two bytes? When encoded into two bytes, 840 bytes would be saved, representing about 3.5 percent of the total bytes used.

One approach is to build a table of the commonly used words in a file. Each word is then assigned a single- or double-byte code value. These code values are used in place of the word, saving a great deal of space. This is the method used in Microsoft BASIC, which "tokenizes" the BASIC statements into (generally) one-byte codes to dramatically reduce storage requirements of BASIC programs.

In an early computer book, *Digital Computer Primer* (McGraw Hill, 1959), Dr. E. M. McCormick, now at California State University, Fullerton, did an interesting analysis on this topic. His entire book was represented on punched cards! Analysis of the 57,936 words in the book showed the most frequent 15 words used (the, of, a, in, is, to, and, be, this, for, are, or, it, as, and computer) were repeated from about 500 to 4,500 times. If these words alone were encoded into two bytes, the savings would be 47,871 bytes or roughly 13 percent of the disk storage.

If the 3,128 most common words were encoded in this fashion, about a 34 percent reduction in space could be accomplished. This suggests that an encoding by letter frequency *and* word frequency results in a space compression of about 47 percent, reducing text disk files to about half of their original size. Even with cheap memory and disk storage, this is a significant amount.

We'll show you what we came up with in the second part of this interesting problem next month. PCM

RUD's REVIEW™
A Guide to Mutual Funds

Software that can easily pay for itself many times over. Use **RUD's** to learn about the mutual fund industry or use **RUD's** to help identify funds meeting your investment needs and objectives. A complete mutual fund information retrieval system on diskette. **RUD's** contains information on over 350 major mutual funds including performance history, investment objective, load, IRA accounts, and size.

- Over 350 major funds
- 20 key features
- 5 year ROI history
- Addresses
- Tandy 1000/1200/3000
- Easy to use and menu driven
- Rapid search and retrieval
- Disk based help files
- Toll free phone numbers
- Informative 32 page manual

"A welcome gift" —*Wall Street Micro Investor*
"A tremendous asset" —*PC Ads*

Only $39.00 + $3.00 shipping

Visual Information Sytems, Inc.
PO Box 42106, Washington, DC 20015
800 638-2000 (301 949-1551 in MD)
VISA & MasterCard accepted
PCM

Submitting Material To PCM

Contributions to PCM are welcome from everyone. We like to run a variety of programs that will be useful/helpful/fun for other Tandy Portable and MS-DOS computer owners. We now support the Tandy portable models 100, 200 and 600 and the Tandy 1000, 1200, 2000 and 3000 MS-DOS computers.

Program submissions must be on tape or disk, and it is best to make several saves, at least one of them in ASCII format. We're sorry, but we do not have time to key in programs. All programs should be supported by some editorial commentary explaining how the program works. Generally, we're much more interested in how your submission works and runs than how you developed it. Programs should be learning experiences.

Pay for submissions is based on a number of criteria. The rate of remuneration will be established and agreed upon prior to publication.

For the benefit of those who wish more detailed information on making submissions, please send an SASE to: Submissions Editor, PCM, The Falsoft Building, P.O. Box 385, Prospect, KY 40059. We will send you comprehensive guidelines.

Please do not submit programs or articles currently submitted to another publication.

If you feel qualified to review software and/or hardware products for computers covered in PCM, send us your name, address and phone number; we will send you a questionnaire form and a copy of our reviewer guidelines.

Programmer's Toolkit: 'Nuts and Bolts' for Programming

Software 1000/1200/2000/3000

An electrician needs a toolkit to do his job. Programmers need toolkits too. The tools of a programmer's trade are interpreters, compilers, code creation tools and special utilities to help debug and document his works of art. The most common utilities needed are source deck editing, listing and cross-reference tools. For those who compile BASIC programs, it's helpful to have a tool to remove unnecessary line numbers. Removal of these unnecessary lines reduces code bulk and makes following the logic easier.

The *BASIC Programmer's Toolkit* Version 1.10 provides three of these tools, plus a fourth on one non-protected diskette. The fourth tool is a line number re-inserter for compiled BASIC source decks. Each tool is a separate program. The utilities are advertised to work on Tandy 1000, 1200, 2000 and 3000 computers.

Lister — A Program Lister

Have you ever LLISTed a program, separated and punched the listing, inserted it into a binder, and discovered the three hole punches removed parts of line numbers, some lines of coding were printed on the perforations and ended up half on one page and half on the next, the pages were unnumbered and the program name didn't appear anywhere?

Lister solves these problems by allowing you to obtain a listing with specified margins, blank lines at the bottom of the page and single or multiple line spacings. It can also list many programs in one program run. A final feature is the ability to list selected pages. Programs must be in ASCII format for *Lister* to function.

Unfortunately, the program did not work with my Radio Shack DMP-120 printer. The program first sent some kind of control code that put my printer

in the graphics mode. All subsequent lines were blank, except the form feed advanced the printer one line. A call to the author revealed that the program was tested on numerous printers and worked successfully. However, most of the printers tested were Epson-compatible. The manual says nothing about compatible printer types.

I was able to get a printout by switching off the printer and quickly switching it on when the program started to run. This chops off the title line and first page number. Next, my printer indented correctly but printed a character at the front of each line, which means "I don't understand the character you sent." There is apparently another control code sent at the beginning of each line. I later tested this program with an Epson printer and it functioned properly. The author has agreed to fix these bugs to permit future operation with a wider variety of printers.

Documentation is adequate and almost unnecessary. The program leads the user through the required input.

Crossref — A Cross-Reference Tool

The critics of BASIC say "spaghetti code" whenever they find fault. It would be nice to reply, "Nonsense, I know where every GOTO goes to and on what lines all my variables appear," but you'd stay up several nights producing such a variable and line number cross-reference. Obtaining this cross-reference documentation is a job for a computer. Fortunately, programs exist that can untangle spaghetti code and produce this useful information.

Crossref is the *Toolkit*'s utility program that performs this task. It also permits a listing of the program. Both types of output can only go to the printer. The program to be cross-referenced must be in ASCII format. Up to 10 programs may be cross-referenced in one run.

First, I ran a small program through *Crossref*. It worked fine. Each line number referenced by a GOTO or GOSUB is printed, followed by a listing of lines that called it. Next, each variable is printed and followed by the lines in which it appears.

After this preliminary test, a 600-line BASIC program was run through *Crossref* to perform an acid test. The results were not too good. The program was not able to properly construct a variable cross-reference listing. Variables that contained BASIC reserve words confused the program. For example, the variable CLINE was listed under 'C'. The program thought the LINE portion was a BASIC command. This type of problem can easily occur when one uses BASIC string functions to decode a line.

One must also build in the logic to determine what constitutes a variable and what constitutes a BASIC command. For GW-BASIC, a variable can be more than 100 characters long. It is the spaces that delineate variables. Other BASICs (e.g., Microsoft BASIC as implemented on the Tandy Color Computer) only allow up to two characters per variable name. Spaces are not needed. Therefore, the logic to recognize a variable must be different for different implementations of BASIC. Discussions with the author revealed that this program was not originally written for GW-BASIC.

The list option of *Crossref* worked reasonably well. I did have a few minor problems. Lines where tabs have been inserted to space the coding inward confused the program. Extra blank lines were printed. The pagination was confused from there on. I was not able to break out of the program during a listing.

Documentation was sufficient and not really necessary. The program asks for all the needed data.

The program author agreed that these bugs would be fixed in a future version.

Number and *Replace* — Line Number Deletion and Insertion Utilities

These two utilities are aimed at the BASIC programmer who compiles programs. BASIC compilers produce faster running code than the BASIC interpreter supplied with most PCs. Generally, the compilers only need line numbers if a line is the object of a GOTO or GOSUB. Removal of the unnecessary lines produces smaller source and object code. It also improves readability. These codes would generally be used in the following scenario.

A program is first written in interpreted BASIC. (In my own experience debug and program development is three to 100 times faster when working with an interpreter.) Before compilation, the program is run through *Number* and unnecessary line numbers are removed. The code is then compiled. If all goes well, the compiled program runs successfully. If sometime in the future the programmer wants to add something else or change a large portion of the coding, he could directly edit the source deck and go through successive compilations and program runs to debug the coding.

The *Replace* code gives the programmer another more efficient way to program and debug the changes. If the programmer first runs his code through *Replace*, it reinserts line numbers. The resultant code then functions with the BASIC interpreter. This later course usually speeds up program alteration and debugging. Note: *Replace* does not reproduce the original line numbers, so all old cross-reference listings would not apply.

The *Number* code could also be useful to those who only program in interpreted BASIC. I found it was much easier to follow program logic flow without all the unneeded line numbers. Of course, you could not run the resultant code with the interpreter.

To test these two codes, both the simple test problem and the 600 line problem were used. First, the programs were run through *Number* to remove unreferenced line numbers. Then the output from *Number* was entered through *Replace*. The source decks coming from *Number* were then run in BASIC and produced correct results. Both utilities worked flawlessly.

Documentation was sufficent and not really necessary.

Conclusions

The cost of the *BASIC Programmer's Toolkit* is moderate. It functions about as well as one can expect from a $10 code package — assuming the author fixes the bugs. *Crossref* is not competitive. There exist a number of public domain programs that produce cross-reference listings with much more detail — where all constants, strings and BASIC reserve words also appear. More expensive cross-reference tools even point out where a variable is defined (set to a value). Personally, I found it a pain to have to convert a code from BASIC tokenized form to ASCII before using *Lister* and *Crossref*. *Number* and *Replace* function flawlessly. Typically, compilers use ASCII code input, therefore there are no added steps required for use of *Number* and *Replace*.

Should you buy them? It depends. If you use interpreted BASIC only, then shop elsewhere. If you need all four tools, the *Toolkit* is a reasonably priced package.

(MVP Software, 1035 Dallas SE, Grand Rapids, MI 49507, $39.95)

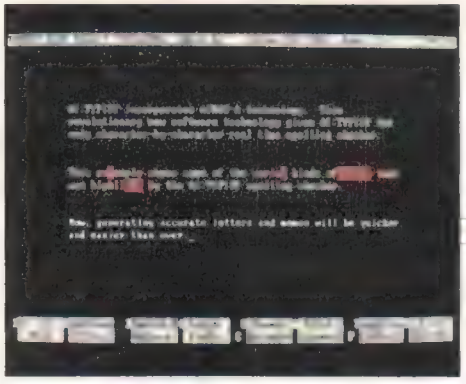
— Brian H. Alsop

"Every once in awhile there is a software product that gives you faith in the future of software."
March-86 AI TRENDS review of AI:TYPIST

"BEEP"

AI:TYPIST 1.16

The only word processing program with a REAL real-time spell checker that highlights any errors you make...without that annoying "BEEP"!



AI:Typist revision 1.16 contains all the features you need in a word processing program. Margin & tabs, cut & paste, search and replace, automatic paragraph reformatting, bold, underline and center text.

But a truly unique feature of AI:Typist is the real-time spell checker. If you make a typing or spelling error AI:Typist will point it out the instant you touch the wrong key, highlighting it on the screen. No annoying "BEEP". You can correct it now or later, and if you really want to be "BEEPED" at, it'll do that too.

AI:Typist, if you make a mistake, it won't tell the world.

SPECIAL OFFER!

$69.95

Order AI:Typist now and save $30.00. AI:Typist has a full 30 day money back guarantee and is not copy protected. List price $99.95. Offer expires 7-31-86

To order call toll free
1-800-547-3000 Dept.222
In Oregon or outside the U
1-503-684-3000 Dept.222

AI:Typist runs on IBM™ and compatible PC with 256KB and DOS 2.0 or later

Accessory 100/200

Power-100 and Power-200 Keeps Your Portable Running

The Power-100 and Power-200 rechargeable battery packs from A.R.M.S. Computer Products are a practical alternative to expensive AA alkaline batteries for your Model 100, Model 200, and various other 6-volt peripheral devices.

The Power-100 weighs five ounces and is ½ inch in diameter. The Power-200 weighs seven ounces and is ⅝ inch in diameter. Both battery packs are 10 inches long.

The Power-100 will provide a Model 100 with 10 hours of power and a Model 200 with eight hours of power. The Power-200 has 33 percent more battery life than the Power-100 and will provide a Model 100 with 13 hours of power and a Model 200 with 11 hours of power.

Both packs easily attach to the back of the computer with self-adhesive velcro. The computer fits into the original carrying case with the battery pack attached.

The batteries connect to the computer where the AC adapter normally connects. The standard AC adapter which powers your computer is used to recharge the battery packs and they can be recharged while you are using the computer thanks to a separate female connector built into the battery pack. This means no down time while charging. A 50 percent charge takes about four hours, while a full charge requires 16 hours. Of course, a 100 percent charge cannot be reached if the batteries are charged while the computer is being used.

(A.R.M.S. Computer Products and Services, 12131 Old Buckingham Road, Midlothian, VA 23113, (804) 794-6675, Power-100: $39.95, Power-200: $49.95, shipping is included.)

— Chris Wehner

Software 1000/1200/3000

Turbo Pascal 3.0: Even Faster and Better

If you're accustomed to programming in BASIC and have shied away from making the transition to PASCAL, it's understandable. Despite its highly-touted structured environment and increased program execution speed, standard PASCAL lacks the rich and diverse repertoire of functions one finds in GW-BASIC.

The object code produced by PASCAL compilers, though it runs faster than BASIC programs, is usually not as fast as object code produced by assembly language or C. Hence, for years, if I absolutely needed fast program execution, I would resort to assembly language and/or C. If speed was not that important and I wanted to complete a software project as soon as possible, there was always the old standby, BASIC.

"...The best software product of its kind, that I have come in contact with." *Computer Language Magazine*

"I've found the best in CCSM...fast in development and fast in execution...no data-typing problems, no concerns for program size, no concerns for file or device opens..." *R.D. Ashworth, Ph.D.*

"...5 years in Basic, Pascal, C, dBase, and Dataflex...I have never worked with a language/programming environment as responsive, easy to use and as powerful as COMP Computing Standard MUMPS" *P.K. Wayne, MD, Ph.D.*

Solve your database problems with CCSM, the Database Language. It comes with a 250 page manual, to lead you step-by-step through this versatile and easy to learn language. Included is a 100 page Introduction to **MUMPS**, presented in an easy-to-follow format.

These Options Will Give You an Even Faster Start!

"Cookbook of MUMPS" 180 page manual with accompanying disk. Includes dozens of fully documented routines and utilities, with examples of controlling output and display, global design hints, and mathematical functions. *RECOMMENDED*

Toolkit I: Pull-down and pop-up menus, pop-up calculator, general-purpose menu driver, screen planning utilities, pop-up notepad, standardized input handler, and demonstration software. *VERY USEFUL*

Multi-tasking, Too! Run multiple concurrent background processes for data searches, report generation, etc.

Also Available **•Multi-User •Graphics**

Equipment: IBM-PC, AT, XT and most compatibles including those by Tandy, Compaq, Texas Instruments, Sperry, AT&T, Kaypro and others. ***Memory:*** *Single User: 128K; Multi-User: 256K and up*

call now for faster service **1-800-257-8052** In Texas 713-529-2576

If you prefer, order by mail

MGlobal
1601 Westheimer, Suite 201
Houston, TX 77006

Please send me the following...

Single User disk and operations manual	$59.95
"Cookbook of MUMPS" (includes disk)	$24.95
SPECIAL Single user & "Cookbook"	$75.90
Multi-tasking	$149.95
Programmer's Toolkit I	$49.95
Graphics option	$49.95
Multi-user	$450.00
Mac**MUMPS** (Macintosh version)	$89.95
shipping and handling	$3.00

Texas residents add 6 1/8% sales tax

VISA__ MC__ AMEX__ expires ____/____/____

card no. _____

name _____

street _____

city _____ *st*____ *zip*_____

Disks are non-copy-protected

day phone_____

This situation changed when Borland International released Version 2.0 of *Turbo* PASCAL. This venerated compiler is the most popular PASCAL for IBM PC compatible and CP/M computers. *Turbo* PASCAL 2.0 required less memory than other PASCALs, took less time compiling, produced programs that ran like the proverbial jack rabbit, had a list of functions comparable to the best advanced BASIC interpreters and cost $49.95!

Now there is *Turbo* PASCAL 3.0.

First, Borland has improved its product so the program object code it creates is even faster. Speeds can actually approach those of hand-assembled machine code. For instance, it took *hours* to sort 1,045 records of 117 bytes each using a BASIC sort program on my Tandy 1200HD. The same algorithm written in Turbo PASCAL took less than *one minute* to sort the same file.

Next, they added more instructions to the language, most noticeably in the graphics area. But they still weren't content implementing PASCAL equivalents of many of GW-BASIC graphics statements (`CIRCLE`, `GET`, `PUT`, etc.). They threw in the convenient "turtle" graphics instructions one normally finds in LOGO for good measure.

There are all kinds of string and screen output functions that evidently exist to facilitate the development of word processors and/or text editor software. Writing a word processing program in *Turbo* PASCAL would be much easier than doing so with BASIC or another PASCAL. Two of the screen output statements are: `InsLine`, which moves lines (beginning with the line the cursor is on and ending at the bottom of the screen) down one line, leaving a blank line at the cursor position; the opposite function, `DelLine`, which deletes the line containing the cursor, then moves all the lines below the cursor up one line.

Turbo PASCAL comes with a program editor. The editor is a *WordStar* workalike. Users who do a lot of word processing with *WordStar* will feel right at home with this editor. The great thing about the editor is that it's interactive, that is, should the compiler find an error, it not only informs the user what and where the error is, but automatically puts the user back into the editor at the problem point.

Unlike Microsoft's MS-PASCAL, *Turbo* PASCAL is a one-step compiler. No linking is required, making it much easier to use, especially for the novice. It also creates more compact code than the Microsoft product, which results in the faster run times mentioned earlier.

The documentation included is a hefty 376 pages. The author assumes the reader has some prior knowledge of PASCAL, otherwise, the material is presented in a fairly straightforward manner.

Some small sample programs are included on the one disk that comes in the package. They're not very sophisticated and really don't do the compiler justice. Perhaps I'm being too harsh. After all, they were thrown in for free.

There are far too many features in *Turbo* PASCAL 3.0 to describe here. But at only $20 more than Version 2.0, it's a steal.

Alas, poor BASIC, I knew it well.

(Borland International, 4585 Scotts Valley Drive, Scotts Valley, CA 95066, $69.95)

— Rick Boozer

YOU ALREADY HAVE HALF OF THE BEST WORD PROCESSING SYSTEM IN THE WORLD

For $49.95* more, your MultiMate, Display Write, WordStar, pfs:WRITE, VolksWriter, Writing Assistant or other WP Software[1] can instantly SET UP TO fourteen columns of text, automatically create outlines, calendars, newsletters, contracts, easy mail merges, screenplays, scripts, footnotes[2], boxed copy and lots more. It's fun with

PowerText FORMATTER

1 Any other MS-DOS or ASCII WP Software
2 On the same page or end of your document

TO ORDER, CALL OR WRITE: BEAMAN PORTER, INC.
417 HALSTEAD AVENUE
HARRISON, NY 10528
(800) 431-0007
(914) 835-3156 IN NEW YORK

*Add $5 for Shipping and Handling.

For Your MSDOS Computer
Tandy 1000/1200/2000/3000 and PCs

RAMDISK ($49)

Use spare memory to simulate one or more super-fast disks. Any size. Greatly speed up existing applications. Appears to programs as a disk, but gives instant access. Highly efficient, compact, and thoroughly tested. Includes program to install and change RAMDISK size, and RAMDISK I/O access report.

FULLBACK ($88)

A Perfect File Backup System !!!

Finally, an easy to use backup program that keeps exact images of your files on backup floppies, cartridges, or hard disk. Automatically backup one, several, or ALL subdirectories. Backup modified only, or ALL files. Keeps perfectly organized backups – backup structure and file format are identical to original. Supports backup by date, multiple backup copies, large files (up to 32MB). Far superior to DOS BACKUP, easier to use, and much more reliable. Absolutely a MUST for hard disk users.

HIP-POCKET HELPERS ()

The ALPS Hip-Pocket Helpers is a collection of powerful and easy to use utility programs. They include a slick Directory Tree, a sorted 5-across directory display, a sophisticated keyboard substitution aid (only 5K of RAM), utilities to search through ALL directories for file names or even strings within a file, and much, much more. 29 utilities in all. Call or write for info. Price and performance you can't find elsewhere.

ALL ALPS Software is Fully Guaranteed !!!
Full Customer Support on our Hotline – 303-687-1442
Call or Write for our complete catalog of products.

ALPS
1502 County Road 25
Woodland Park, Colorado 80866
800-232-ALPS (Toll-Free)

Software 100

Business Decisions Helps Take Care of Business

A series of six software programs called *Business Decisions* have been packaged for the Model 100 by Tandy for the business and finance world. The programs cover a variety of business decisions including inventory replenishment, sequential decision analysis, financial ratios, break-even analyses, linear programming and pricing tables.

The software requires a Model 100 with at least 24K of memory, a cassette recorder and an optional printer. The programs are supplied on cassette and written in BASIC so backups may be made. The program uses an arrow display to allow the output from all the programs to go to the printer or display.

The first program is the inventory replenishment analysis. It allows one to model the company's ideal inventory levels, reorder levels and frequency of reorders for maximum savings. To run you need the following information: annual unit demand, product unit price, single order cost, unit holding cost, shortage costs, capacity of storage and maximum inventory capacity. The output shows the optimum order, optimum inventory, price per unit, holding cost, ordering frequency and limited capacity cost. This is a thoroughly thought out way of presenting all the information.

The sequential decision tree analysis is a "what if" tree of events. If one thing happens, what is the end result? It really evaluates the probabilities involved in risk taking. An example of decision tree analysis would entail a scene as such: A factory decides to build a new appliance. It will cost a certain amount to tool up, a certain amount to advertise, a certain amount to get rid of the material if it doesn't sell. The probability of it selling at a price is some percent, the probability of another manufacturer making it cheaper or better is a certain probability, along with the probability of the weather. All these items comprise a sequential tree. This program determines the overall success of each determinant and the project.

The third program determines the financial ratios such as liquidity, leverage, management and profitability. The liquidity ratio shows the company's ability to pay its short term obligations. It is based upon the current assets and liabilities and inventory. The management ratios give information about the inventory turnover, average collection time, fixed asset turnover and total asset turnover. The program also calculates the net profit margin and return on assets after tax profits ratios.

The break-even analysis uses the fixed costs, variable costs, unit price, maximum production and projected sales volumes to determine the break-even point.

The most complicated program is the linear programming. You have the opportunity to enter production limits, labor costs and labor volume for many different items, i.e., brands of colas. Each brand will have certain constraints and this software wades through the muddle and gives a mathematical equation that is evaluated by the computer to determine, for instance, how much of each cola should be produced to give the maximum plant operating efficiency.

The last program is the pricing table. If you have several items to sell and each has a verifiable cost, then it will calculate the selling price and possible discount prices if you can give quantity discount. It optimizes the price based upon selling volume and cost.

The manual accompanying the programs is excellent in the true Tandy fashion. The programs are explained along with what data is necessary, and examples with printouts of actual data are given. The formulas used in the programs are given in a separate chapter. All the information you need to be up and running is included.

If your business needs require software included in this selection, I recommend you take a closer look at this latest offering from Tandy.

(Tandy Corp., Fort Worth, TX 76102 $39.95)

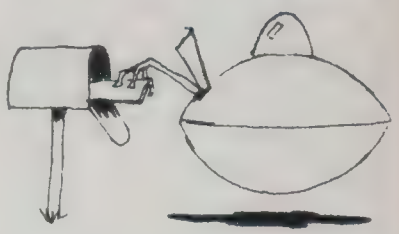
— Vincent Lord

Software 1000/1200/3000

Take a Different 'Approach' with *Solo Flight*

From this pilot's viewpoint, the most exciting thing about *Solo Flight* is the letter from Bill Stealey, Micro Prose Software President, which is printed in the operation manual that comes with the software. The letter is written with the enthusiasm that pilots feel when they talk about flying. After the build-up, "most fun, most like flying, most

PCM



Back Issue Availability

Back copies of many issues of PCM are still available.

All back issues sell for the single issue cover price. In addition, there is a $2 charge for the first issue plus 50 cents for each additional issue shipped in the U.S. When possible, issues are shipped UPS. The postage cost in Canada and Mexico is $3 for the first issue and $1 for each additional issue.

VISA, MasterCard and American Express accepted. Kentucky residents please add 5 percent state sales tax. We do not bill and no COD orders accepted.

We suggest you order back issues you want now while supplies last.

To order, just fill out the form on the next page and mail it with your payment to:

PCM
The Falsoft Building
Prospect, KY 40059

PCM

☐ Please send me the following back issues:

(Payment must accompany back issue orders.)

NO.	MO.	YR.		PRICE	
			VOLUME 1		
1	JULY	'83	PREMIER ISSUE	$3.00	☐
2	AUG.	'83		$3.00	☐
3	SEPT.	'83		$3.00	☐
4	OCT.	'83		$3.00	☐
5	NOV.	'83		$3.00	☐
6	DEC.	'83		$3.00	☐
7	JAN.	'84		$3.00	☐
8	FEB.	'84		$3.00	☐
9	MAR.	'84		$3.00	☐
10	APR.	'84	LIMITED	$3.00	☐
12	JUNE	'84	*	$3.00	☐
			VOLUME 2		
2	AUG.	'84		$3.00	☐
3	SEPT.	'84		$3.00	☐
6	DEC.	'84		$3.00	☐
7	JAN.	'85	**	$3.00	☐
8	FEB.	'85	***	$3.00	☐
9	MAR.	'85		$3.00	☐
10	APR.	'85		$3.00	☐
11	MAY	'85		$3.00	☐
12	JUNE	'85		$3.00	☐
			VOLUME 3		
1	JULY	'85		$3.00	☐
2	AUG.	'85		$3.00	☐
4	OCT.	'85		$3.00	☐
5	NOV.	'85		$3.00	☐
6	DEC.	'85	****	$3.00	☐
7	JAN.	'86		$3.00	☐
8	FEB.	'86		$3.00	☐
9	MAR.	'86		$3.00	☐
10	APR.	'86		$3.00	☐
11	MAY	'86		$3.00	☐
12	JUNE	'86		$3.00	☐
			VOLUME 4		
1	JULY	'86		$3.00	☐

* Tandy 2000 coverage begins
** Tandy 200 coverage begins
*** Tandy 1000 coverage begins
**** Tandy 600/3000 coverage begins

TOTAL _____

KY RESIDENTS ADD 5% _____

SHIPPING & HANDLING _____

TOTAL AMOUNT ENCLOSED _____

NAME _____ ADDRESS _____ CITY _____ STATE _____ ZIP _____ ☐ PAYMENT ENCLOSED

☐ VISA ☐ MC CARD # _____ EXPIRATION DATE _____ PHONE # _____

SIGNATURE _____

tested by real pilots," *Solo Flight* was a real let down. But, don't despair, flying buffs, this review has a happy ending.

It started off like my kind of program: except for the fact that the manual does not include instructions specifically for the Tandy 1000, it appeared to be adequate and easy to follow. The program booted up with no special installation steps. Choices to be made were menu driven and logical. Then the disappointments began. I reluctantly adjusted to the lack of numbers on the instruments, but never did get used to north on the compass looking like an 'H' instead of an 'N'.

The fact that I had to go from 20-degree flaps to 40-degree flaps in order to get to 0-degrees bothered me. The VOR indications left much to be desired. The joystick response was incredibly slow. But it was the fixed horizon that robbed any chance at reality.

If you are a certified pilot looking for a realistic flight simulator to play with on rainy days, this is not for you. You get about as much of the "feel" of flying as you would watching a plane fly overhead. That's largely because of the fixed horizon. When you are really in the pilot's seat and those wheels leave the ground, that horizon begins to teeter back and forth. You turn left, it tilts right. You put the plane into a steep climb and the horizon disappears below you.

In *Solo Flight*, you watch the plane you are controlling much as you would a remote-controlled plane. And in that comparison comes some complimentary evaluation. Forget that the publishers said anything about realism or the feel of flight and *Solo Flight* can be a fun flying game.

After choosing Flying Practice, you get a choice of flying in six different states with varying difficulties in terrain and weather — Kansas being one of the easier states, and Colorado, one of the more difficult. Then you must choose the condition you want: clear weather, landing, windy, IFR or night. If you choose clear weather flying in Kansas, you will have an instructor on board. Commands are written on the screen, or you can get rid of the instructor by simply typing I. The instructor feature is a good way to learn to fly your simulator airplane.

Pilots should have no trouble understanding the manual. I'm afraid anyone who does not have the knowledge of real flying will experience some difficulty with it. Also, in the program itself, the indications on the compass are 'N', 'S', 'E' and 'W'. The on-board instructor gives the command to turn to 270-degrees. Assuming everyone knows the cardinal directions of the compass is taking a lot for granted. The manual and the instructor's commands are in a pilot's language, but the program is not what I would call a real pilot's program.

After flying around in a state the program goes to a map of that state and shows where you've been by plotting the path you have flown. I like this feature very much. There is a map for each state printed in the manual to assist you in planning a trip. Each state has two VORs (navigational aids) to help you get where you want to go. Airports are easy to spot; making an acceptable approach is a little more difficult. The landing practice option gives you the opportunity to perfect your approaches and touchdowns. In my opinion, a real plane is considerably easier to land.

"If you are a certified pilot looking for a realistic flight simulator to play with on rainy days, this is not for you."

The other choice in the program is Flying Mail. Here's where you can really have some fun and use the skills you've developed in Flying Practice. After choosing the state and one of four levels of difficulty, you load the mail and fuel. Points are scored based on the state you choose, time it takes to fly the route, the level of difficulty chosen and your skill in landing. Time between airports is shortened and this is a plus factor in making this game interesting. The game is challenging and fun, and as your skills improve, you can make it more challenging.

Here again, however, is where the manual leaves something to be desired. In giving instructions for loading the mail and fuel, it reads "You may load

up as much mail as you like. However, each bag adds to the weight of your plane and increases the difficulty of flying. One or two bags is recommended." It goes on to say, "Now choose the amount of fuel you want to carry. Remember fuel adds weight. The more weight the more sluggishly the aircraft responds. On a hot day, at high altitudes, in Colorado you may not be able to get off the ground!"

While a real pilot understands the importance of weight and balance, I doubt if a novice would understand the implications of these instructions. If the player takes time to consult the "Specifications and Performance" chart, he'll find his fuel capacity is 24 gallons, but the mail game allows him to load 64 gallons of fuel. This is somewhat confusing. There is no mention of long-range tanks or a different aircraft for the mail game.

With 64 gallons of fuel and six mail bags, I was about double the maximum gross weight of the aircraft. While unable to lift off at Aspen, Colorado, I did make a successful takeoff at Wichita, Kansas using 20-degree flaps. I climbed through the 10,000 feet overcast, and experienced no real problem until I began an extended, 75-degree power climb that caused my engine to overheat. I came back down through the overcast, located an airport at Lyons, Kansas, and made my approach to land.

"As your skills improve, you can choose greater levels of difficulty in several ways. It should take a long time to really become a master at this game."

In spite of the inconsistencies in the manual and the game, I like the Flying Mail feature very much. A rewritten manual with more detail and a better explanation of some of the features of the program would be a great improvement. The thing that bothers me most about *Solo Flight* is the promotional material, which makes such a point of calling the Simulation realistic. The manual is written with an obvious knowledge of flying and is realistic in what it says, but the program itself lacks that realism.

Solo Flight would have scored much higher with me if its publishers had allowed it to stand on its own. It is fun and challenging; if you are a flying buff, it's a program you'd have a hard time becoming bored with. As your skills improve, you can choose greater levels of difficulty in several ways. It should take a long time to really become a master at this game. At $34.95 I would consider it a pretty good buy.

(Micro Prose Simulation Software, 120 Lakefront Drive, Hunt Valley, MD 21030, 301-667-1151, $34.95)

— Irma S. Canfield

Software **1000/1200/3000**

Shorthand Goes High-Tech with *Productivity Plus*

Although there is some indication programs that perform somewhat similar tasks may appear on the market soon, *Productivity Plus* has the advantage of being here first. Besides, since it is a superior program it stands to hold its own with any competition, if not to beat it out altogether.

Boiling it down to the simplest terms, *PRD+* (as its creators call it) is an easy-to-use, menu-driven program that enables you to use alphabetical shorthand on the computer keyboard. You can use the basic word list provided with the program, add to the list or create personalized short form/long form word lists that, when invoked, automatically bring to the screen almost any intended long form you may choose to create. The personalized lists can be temporary or permanent, they may be created while in the midst of writing text, or separately from a list creation provision in the program that is easily invoked as part of the main menu.

It works like this: Suppose there is a phrase you use often, such as "for your information." This phrase is one of those included in the basic list of short form/long form combinations. The letters "fyi" constitute its short form. Suppose you are writing a letter in which you wish to say, "The enclosed press kit is for your information." Using *PRD+*, all you need to type is "The enclosed press kit is fyi." To shorten things further, you could arrange it so all you have to type is "thpc z fyi"; "thpc" standing for "The enclosed press kit" and 'z' for "is." If you sent out press kits frequently, the saving in keystrokes is obvious.

Pressing the period, or any other delimiter for that matter, causes each of the long forms of "thpc z" and "fyi" to print out on the screen at the touch of a single key. Therein lies the principle advantage of *PRD+*. Once loaded, it resides in RAM. When it is toggled on, any delimiter causes any long form that is on a word list to print out on the screen. Word lists may be used by themselves or merged with other word lists that are added to the program disk according your personal needs.

The lists that come with the basic program contain a substantial number of commonly used words and phrases. One of the handiest parts is the complete list of codes for all 50 states. To print out the full name of any state when writing an address, just type the two-letter postal abbreviation in lowercase, press the delimiter you require and the name of the state prints out properly capitalized. A number of major cities are also on the list.

The on-screen tutorial is excellent and the manual is not only unique but one of the best I have seen in terms of clarity, simplicity and ease of understanding. It is a ring-bound steno pad in size and shape, but it has a folding base that allows you to stand it up alongside your monitor — a big help when familiarizing yourself with its details and even more so when you want to have the word lists handy for reference while you work. There is also ample space on every page to make your own notes.

PRD+ was designed for the busy user who wishes to save time and does not want to be bothered with a lengthy learning process. Installation is simple and the program can be customized for most popular word processors. It can be used with *dBASE, Lotus* and virtually any database program or spreadsheet; it can be used to store complex mathematical formulae as well as text. The latest Version 1.11 includes a macro storage feature. Short forms can be as short as a single character (though this is not recommended) and long forms up to 240 characters. When you can store a frequently used phone number or address, for example, and call it up by

typing three characters and a space, you wonder how you ever did without such a timesaver.

On the subject of timesaving, there is also a feature that enables you to check on yourself and see how many keystrokes you have saved in the creation of any given document.

PRD+ is not copy-protected, it takes up only 64K of RAM and requires MS-DOS 2.0 or better on any IBM PC, XT/AT and compatibles. Interestingly enough, when we tested *PRD+* we found that on the Tandy 1000 it causes the keyboard to be slightly redefined: the cursor keys assume the positions they normally occupy on IBM-type keyboards, the '2', '4', '6' and '8' keys respectively. The '7' key on the numeric keypad becomes the HOME key and, when unshifted, the PRINT key becomes an asterisk. The left cursor becomes a backslash, the right cursor a plus sign, the up cursor an inverted apostrophe, the down cursor a dash, and zero on the numeric keypad, INSERT. Though it sounds confusing at first, one user who tested the program said it was essentially the same as using a program like *Prokey* or *Superkey*, and that the advantages of *PRD+* more than outweigh the minor matter of getting accustomed to the slight keyboard redefinition.

More over, in some instances, it proves to be a decided advantage. The word processor, *Office Writer*, an excellent Wang emulator, cannot be used on the Tandy 1000 because it requires the '+' key at the right side of the numeric keypad to be used to enter commands. When redefined by *PRD+*, the former right cursor key becomes a '+', allowing *Officer Writer* to be run. With the database *InfoScope*, the same key becomes a power key that speeds up cursor movements under certain desired circumstances, another advantage.

In short, *Productivity Plus* is a superior piece of software on all counts. It does precisely what it is supposed to do; it is easy to learn. Even the most inexperienced user can learn it in about 30 minutes. It is a secretary's dream, especially those who are familiar with alphabetical shorthand such as speedwriting and stenoscript. It is ideal for the office that uses PCs. Organizations using specialized terminology will find *PRD+* a boon, especially in this age when even the most educated computer users often spell badly. Once a frequently misspelled word is given a simple short form and saved on a word list, it will never be misspelled again. That in itself can be worth the price of admission.

(Productivity Software International, Inc., 1220 Broadway, New York, NY 10001, 212-967-0947, $195)

— Bernhardt J. Hurwood

Software **1000/1200/2000/3000**

Word Finder: The Thesaurus Goes High-Tech

You're pounding away at your word processor and suddenly something doesn't seem right. In the middle of the sentence you just typed is a word, and it isn't the word you really want to use. The "right" word is on the tip of your tongue. What now? Off of the bookshelf

MARYMAC INDUSTRIES INC.

800-231-3680

Radio Shack® Tandy®

EPSON PRINTERS

People you Trust to give you the very best!

- Lowest Discount Prices
- Reliable Service
- Quality Products

"World's largest independent authorized computer Dealer"

22511 Katy Fwy., Katy (Houston) Texas 77450

(713) 392-0747 Telex 774132

FREE!*

WORD PROCESSING TRAINING GUIDE for **IBM-PC** and **TRS-80**

Become an expert in the advanced word processing techniques in just a few hours. Develop skills that can be applied to any of the best selling word processors, no matter which system you decide to buy. Includes program disk so you learn quickly by doing. The training guide and training program disk (a $50.00 value) will be sent to you upon receipt of this filled out coupon without any obligation or cost, *except shipping and handling. Offer good for a limited time only.

LeScript Word Processing Systems
P.O. Box 361136, Melbourne, FL 32936
(or call 1-305-259-9397)

Type of computer system:
☐ IBM-PC/XT/AT/jr compatible (includes Tandy 2000)
☐ TRS-80 1/3/4/4P
☐ TRS-80 4/4P CP/M

Name

Address

City State Zip

Telephone

*Please enclose $2.00 to cover shipping and handling.

(Computer names are tradenames and/or trademarks of their respective manufacturers.)

comes the thesaurus.

Now wait a minute. Don't you feel a little inefficient? If this was a typewriter it might be different. But this is a high-powered word processor. Reaching to the bookshelf is like reaching back to the century of Gutenberg, a little like grabbing a slide rule in the middle of your spreadsheet session. There's got to be a better way!

With *Word Finder* from Writing Consultants there is! Billed as a synonym finder, this computerized thesaurus is memory resident. Simply place the cursor anywhere on the offending word and press the command key. The *Word Finder* window appears displaying the word followed by a list of synonyms. If the "right" word is one of those displayed, simply move the cursor in the window to that word, press RETURN and you are returned to your word processor with the new word in place.

"*Word Finder* is invisible until summoned. The window that appears upon using the command key leaves your text visible on the screen with the target word highlighted."

Word Finder (Version 3.2) lists "PC compatible" as the only hardware requirement. On a dual floppy system, users are advised to create a disk for Drive A with the essential files from their word processor, together with the two essential *Word Finder* files (the executable file and the synonym file). In Drive B goes a disk with their text files. The synonym file in this case includes 90,000 synonyms.

For the hard disk system user, the program includes an expanded synonym file on a second diskette with over 220,000 synonyms. These files are not copy protected, allowing location anywhere on the hard disk. The full power, speed and convenience of *Word Finder* is achieved with a hard disk system. The documentation lists 23 of the most popular word processors as being compatible with *Word Finder*, with a suggestion to try installation with others not listed. Prospective purchasers should probably contact Writing Consultants regarding compatibility. The program performed flawlessly on a Tandy 1200HD running *WordStar*.

My only (minor) criticism of *Word Finder* concerns the documentation. The package I received included a documentation booklet from a previous version, together with a separate update booklet including revisions keyed to the original. Hopefully, the changes will soon be incorporated into revised documentation. In spite of this, the documentation led me easily through the installation procedure, which modifies the executable file to be compatible with the particular word processor.

Word Finder must be loaded prior to the word processor. Typing WF loads the program into memory and returns to DOS. The word processor can then be loaded, leaving *Word Finder* resident. A simple batch file can be created to simplify this process. Once into the word processor, the operation of *Word Finder* is very easy, eliminating the need to refer to documentation.

Word Finder is invisible until summoned. The window that appears upon using the command key leaves your text visible on the screen with the target word highlighted. In the case of *WordStar*, the window takes the place of the menu at the top of the screen. This feature allows you to view the word in context while deciding on a synonym.

Synonyms are presented by part of speech (n=noun, v=verb, etc.). When a word has more than one meaning, synonyms for each meaning are grouped together. The different meaning groups are separated by semicolons. Long lists of synonyms may require multiple windows, accessed with PgUp and PgDn. If just the right word is not among those listed, but one seems closer to the mark, place the cursor on that word and press the command key for a list of its synomyms. This is a different list from the first. This process can be repeated to home in on the desired concept and find the precise word.

Replacement of a word leaves the original form of capitalization and punctuation intact. If you decide your original word really is best, simply press escape to leave it intact. When the window disappears, you are returned to your word processor exactly where you were before calling *Word Finder*.

If the target word is not in the program's dictionary, the window displays this message: The word was not found. Choose another word to look up. Included in the window are the 30 words alphabetically closest to the target word. This feature may help you to zero in on that synonym. It also works well as a real-time spelling checker! *Word Finder* can be used without a word processor, if desired.

I've always struggled when resorting to a thesaurus. It seems to take too long, and all too often I come away without a better word. *Word Finder*, by contrast, is a breeze to use. The program provides easy access to an impressive number of alternative words (especially with the expanded synonym file available to hard disk users), and should assist any writer. I highly recommend this program.

(Writing Consultants, Techniplex Center, 300 Main St., East Rochester, NY 14445, 716-377-0130, $79.95)

— Stanley Townsend

Software **1000/1200/3000**

ProDisk is Truly User Friendly

With so many software companies carelessly bandying about the term "user friendly," many users have come to fear any claim made about friendliness. The Harvey Invisible Software *ProDisk* package may restore our faith in the term. Here is a piece of software that is not only useful, but very user friendly.

Harvey *ProDisk* is a memory resident, windowed-menu system designed to make execution of programs simple and painless. Although the package comes with several other independent programs, the *Menu* system must be considered the heart of the *ProDisk* package. With *ProDisk*, the user can create menus that allow the execution of sophisticated command sequences or

programs with a one- or two-stroke entry.

When a menu (window) is on-screen, any application on the menu can be executed by entering the first letter of the selection, or by using the arrow keys to highlight the selection, then pressing ENTER. Even users thoroughly intimidated by the complex commands required to operate under DOS should quickly find themselves at ease with *ProDisk*. Password protection is offered for each menu to prevent unwelcome users from accessing private information.

Creation and editing of the menus is not difficult at all. Pressing the F5 key brings up the creation screen. Making a new menu is a three-step process. First give the menu a name, then enter each selection and follow it with the commands needed to execute that choice.

"The program claims to be compatible with other off-the-shelf software. I found no problems operating it with *GEM, Sidekick, Turbo Lightning* or any other memory resident programs . . ."

To change a menu entry, the F6 key is pressed. A submenu pops up, offering to edit either the selection, the menu, color or password. Once the choice is made, the new information is entered and overlays the old. Press F8 and the new menu becomes permanent.

Menus can be called from other menus, forward or backward. As an example, suppose you had a menu called Main, from which all major submenus originated. To call Main from one of the other menus, you would include in the other menu a selection for "Exit to Main Menu." For the command entry, you would type /M MAIN. Anytime the 'E' is pressed while in this submenu, the user would be returned to Main menu.

The program claims to be compatible with other off-the-shelf software. I found no problems operating it with *GEM, Sidekick, Turbo Lightning* or any other memory resident programs and I see no reason to doubt compatibility with other programs.

A package of secondary utilities comes with *ProDisk*. These programs, *HTime, HView, HPrompt* and *GF* (Graphics Filter) are all well-written and perform useful tasks.

HTime provides a unique view of a wristwatch on the screen with a current month calendar highlighting the current date. Other time/date related functions can be accessed as well, including such things as calculating the time between two dates.

HView is a replacement for the DOS commands TYPE, PAUSE and MORE (filter). You may *HView* a file one page or one line at a time, paging backwards and forwards as you wish.

The *HPrompt* is an alternative to the standard A> or C> prompt. It includes a rectangular box at the top of the screen with a full-width command line and the current directory. Once invoked, it replaces the standard prompt whether or not the *ProDisk* program is loaded.

GF converts graphics characters to the character given as their parameter.

Do I have anything negative at all to say about the package? Perhaps one minor thing. The documentation is two standard size, quick-reference sheets plus an on-disk tutor that can be printed out. The formatting is fine for on-screen help but is not, in my opinion, right for printing out. Perhaps a separate file for printing out, or preprinted documentation would be better. Fortunately, the on-screen help and ease of program use make this a minor problem at most.

If you are a hard disk user, and particularly if your business allows access by non-computer oriented people or you have private information to be made available only to certain individuals, this program has a lot to offer. Overall, I heartily recommend it to readers of PCM and fellow Tandy MS-DOS users.

(Harvey Invisible Software, P.O. Box 1863, Cape Coral, FL 33910, 800-231-0296, $90)

— Leonard Hyre

Software **1000/1200/3000**

Software Review . . .

AI:Typist — An Easy-to-Use Word Processor

I've always liked programs that can be used right out of the box. The thought of wading through page after page of a manual and having to learn a lot of new commands and procedures has often given me second thoughts about purchasing new programs. *AI:Typist* requires no previous experience to use, just spend a few minutes reading the introductory chapter and you're ready to process words! I started using the program immediately and my 12-year-old daughter was able to type a paper for school with less than 10 minutes of program orientation.

The non-copy-protected program requires 256K RAM and DOS 2.x or higher. Features include; help screens, search and replace, cut and paste, document formatting, multiple printer support, advanced *WordStar*-like commands (just in case you don't like the simple and effective menus), ASCII- and *Typist*-formatted file operations, and a lightning fast, memory resident spelling checker.

The main screen is the heart of the program, the top of the screen displays the document page, line, word count, column, percentage of 64K the document uses, the document name and insert or overwrite status. At the bottom of the screen there is a block display of menu functions for function keys F1 through F8, and flags to indicate the status of the NUM LOCK and CAPS keys, and the automatic paragraph format status.

Key designations are straightforward; INSERT, DELETE, the arrow keys, HOME, NUM LOCK, CAPS, and BACK SPACE all work just like I expect them to. All special functions are activated with the touch of one of the accurately-labeled function keys. F1 is always the "help" key and produces on-line help screens from anywhere in the program. F9 deletes a line and F10 inserts a line.

F2 controls the "get/save" file functions and will insert a document at the cursor position or retrieve a file for editing. Files can be saved in *AI:Typist* format or ASCII format. F3 triggers the search and replace functions; words or phrases can be searched out and replaced throughout the document. Cut and paste functions are controlled by F4; any block of text can be marked, removed, or moved to a new location. The features controlled by F5 allow rapid formatting of the document;

margins and tabs can be set, automatic paragraph adjustment can be toggled on and off, lines can be centered, areas can be marked for bold or underline printing and entire blocks can be reformatted.

The document printing menu (F6), supports 18 standard printers and two custom printers. The entire document can be printed or printing can be selected for one page at a time. There is a provision for double-spaced printing, selection of alternate printer ports and special characters embedded in the text.

The spelling checker is activated or deactivated with F7. The warning beep can be turned off or on and the alternate dictionary is maintained using the "Add to Dic" and "Delete From Dic" functions (the main dictionary cannot be altered). There are provisions to spell check entire documents (any ASCII document can be checked) or simply find the next misspelling in the current document. The spell checker does not offer suggested spellings or automatic correction, it simply flags words that are not in the dictionaries.

The documentation that comes with *AI:Typist* consists of a complete, accurate and easy to read manual with a compact quick reference card. Although well-done, the documentation is really redundant in light of the program's user-friendly construction and design. Fast and friendly customer service is available by calling (503) 246-1782.

AI:Typist is not a replacement for the super-powered, high priced word processing programs. It does not support microjustification, automatic page numbering, on-screen page breaks, right justification, mail merging, variables, or a fancy thesaurus. Only two special printer functions can be used at one time and documents are limited to 64K (20-30 pages).

AI:Typist is ideal for typing letters, memos, school reports, proposals and almost any short- to medium-length document with spell checking that occurs as fast as you can type. There are no fancy commands to memorize or complex procedures to learn, and little or no computer expertise is required. For anyone interested in word processing at a reasonable price, *AI:Typist* is an excellent choice.

(Airus, Inc., 10200 S.W. Nimbus Ave., Suite G-5, Portland, OR 97223, $99.95, 503-246-1782)

— Bob Mills

Software **1000/1200/3000**

Popcorn Desktop — Useful and Affordable

Popcorn Writer is a small, fast word processor that uses codes familiar to anyone who has ever used *WordStar*. In my opinion this gives it three distinct uses.

The beginner who needs a word processor, but can't afford $300 and up for a full-featured product, will be grateful for having learned *Writer* when he needs to use an advanced text editor because most of the *WordStar* commands will already be familiar.

The person who uses *WordStar* at work can afford this home editor and won't have to learn new commands.

Third, I would recommend this to many small businesses because of a nice added feature. *Writer* comes with an easy-to-use Mail List Merge feature for producing mailings.

Writer has one feature missing on *WordStar*, a good Undelete command, but performs most other functions of advanced word processors. There is also a nice on-disk tutorial that introduces the beginner to this kind of word processor.

The *Popcorn Misspeller's Dictionary* is just that, a dictionary, not a spelling checker.

Misspeller lets you produce your own dictionaries and, since it is RAM resident, it is available anytime the computer is on.

There are two sizes of dictionaries supplied in *Misspeller* that you can add to, or your word list can be completely separate.

Why a dictionary when you can buy spelling checkers? First, you won't get a good spelling checker for this price and, second, you may only occasionally need to check the spelling of a word. By using a dictionary when needed instead of a checking program on every word you will save a lot of time.

Popcorn Desktop is also RAM resident and contains a pocket-type Calculator, an ASCII table, a Calendar (for viewing only), on-screen help and a Notepad.

Interestingly, you can install all of these desktop functions at once, or either Notepad or Calculator alone, if you are short on space.

Notepad is the interesting program. It uses *WordStar*-type commands and has lots of useful features; you can call up a directory of any disk drive. This program has all the usual editor features: Word Wrap, Saving, Printer Control Codes, Block Operations, Search and even Undelete, very similar to *Popcorn Writer*.

It also has one terrific feature. Notepad lets you block out any rectangle on the screen and import the information into Notepad. This means you can have a directory, help file or anything else shown on the screen available no matter what program is running. For anyone who fools around with BAT files, or otherwise jumps from program to program (like reviewers), this feature is great. This was a very easy review to write because I could refer to the operating program while writing.

"The programs in *Desktop* and *Misspeller's Dictionary* can all be layered on the screen at once, and the windows they occupy can be moved around."

The programs in *Desktop* and *Misspeller's Dictionary* can all be layered on the screen at once, and the windows they occupy can be moved around (the program remembers where and returns to the new position the next time).

You can move some screen data into Notepad, then write a note, check the spelling with *Dictionary*, look at the Calendar and then do some calculations, with all programs onscreen at the same time. Each ESC key removes another window until you are back in MS-DOS.

All these programs come "preinstalled," that is, you just load them and execute on many machines, but you can also custom-install the programs if you want to (or need to). They all require an MS-DOS computer with 128K of memory, they support color or monochrome monitors, floppy or hard disks, and are not copy-protected.

All three products come with short, clear documentation that will have

anyone using the software in minutes.

These programs share one peculiarity — after invoking one of them and returning to MS-DOS, the next two keystrokes are ignored. This is due to the way Notepad captures data from the screen, and never posed a real problem to me once I discovered it was happening. However, this doesn't happen when typing in the word processor.

(Popcorn Software, P.O. Box 814, Torrance, CA 90508, *Writer* $69.96, *Misspeller's Dictionary* $29.95, *Desktop* $49.95)

— John McCormick

Book **1000/1200**

Learning BASIC For The Tandy 1000/2000 — An Excellent Choice

For those of us who have other Tandy computers, David Lien is a familiar name. He has written books and handbooks for the Model I, Model III, Model 4/4P and the Model 100. Now Tandy has enlisted his considerable talents to teach Tandy 1000/2000 BASIC to another group of computer owners.

I think that Tandy has made an excellent choice. While *Learning Basic For The Tandy 1000/2000* is designed and written for the beginning beginner, the writing style and presentation should hold the attention of even experienced programmers. I have written a number of BASIC programs, but there were several times when I said to myself, "I didn't know that."

When I say written for beginners, this book even tells you where the power switch is located for both models. It instructs the reader on entering time and date as requested by DOS at boot-up. This description includes telling the reader where on the keyboard to find the colon (:). Following that is a diagram of a floppy disk and a description of how to format the disk for use.

In any book of this type, the layout is crucial to following the material. The text is in black ink and regular type. The lines to be typed in by the reader and the responses from the computer are printed in blue. This technique makes it much easier to figure out what you are supposed to type and how you should expect the computer to respond. Keys to be pressed (ENTER, CTRL, etc.) are printed in reverse blocks of white letters on black. Scattered throughout the text between solid horizontal lines are hints, reminders, expanded explanations and a few jokes. These are almost like footnotes, but they are in the text rather than at the bottom of the page. Finally, full-page cartoons occur from time to time to keep things lighthearted.

The flow of the book is logical and each new chapter and example program is built well on what comes before. FOR-NEXT loops are very important and their explanation is particularly good. I will admit, however, to yawning a little in the chapter on precision. I think it was slight overkill in a beginners text, but if you make it through all the material, you will understand single and double precision better than most programmers. PRINT USING print formatting was also covered in great detail, but the examples seemed more useful to me.

Arrays and even sorting are covered

Now . . .
Beyond LORAN C!

Nav Easy
DOES IT!

IDEAL FOR RENTERS

EAGLE AIR SERVICES
1501 East Chapman Avenue
Suite 258
Fullerton, California 92631

EXPAND THE TANDY 2000®
TO
896K
OF DOS ADDRESSABLE RAM
Includes RAMdisk Software at No Extra Cost ...
Add 640K of External RAM Memory to TANDY 2000's® 256K of Internal RAM Memory for a total of 896K of DOS Addressable RAM.
★ ★ New Products Available Soon ★ ★
(Send for catalog)

cost: $495.00 for a 640K bytes of External RAM Memory Board (Includes RAMdisk Software)
Call about our Trade-in Exchange option if you already use Tandy's 128K External Memory Board.

for ordering and information call: (509) 627-5291 Hours 9:00-5:00 PST

terms: We accept Check, MO, Visa, Mastercard
Add 7.3% sales tax in Wash.
Add $3-shipping/handling, $5 foreign.

TANDY 2000® IS A TRADEMARK OF TANDY CORPORATION
RAMdisk is a Random-Access-Memory based disk

ENVISION DESIGNS
1909 ORCHARD WAY • RICHLAND, WASHINGTON 99352

clearly including multidimensional arrays. Sequential data file use is covered adequately, but random file handling is not mentioned at all. This is no surprise since the book promises only sequential files. I liked chapter 49, "A Study of Obscurities." The explanation there of VARPTR is the only one I have ever understood. Chapter 51 is on debugging programs and contains techniques that are an absolute necessity for a beginner.

Enough already. If you have a Tandy 1000 or Tandy 2000 and want to learn how to use the BASIC that comes with it, this book is the one to buy. I would give it five stars out of five for doing the job it is designed to do.

(Author: David A. Lien. Published by Compusoft Publishing. Radio Shack catalog no. 25-1500)

— Potter Orr

Hardware **1000**

A2D Joystick: Quality to Match the 1000's

All right, everyone knows the Tandy 1000 is a business computer. After all, it is I-B-M ('B' as in Business) compatible and runs all those businessy programs like *WordStar, Lotus 1-2-3* and *Harvard Total Project Manager*; nobody pays a quarter to play these in an arcade.

Well then, what are those two ports to the far right of the big red Reset button doing there? They are labeled "joystick." It seems the designers of the 1000 had more in mind than crunching numbers and processing words. It comes standard with color graphics (16 colors no less), sound and the ability to control programs with a joystick.

After an extended session of pushing numbers, it's nice to be able to make friends again with the computer (mine is named Fred). After a few minutes of concentration on nonbusiness pleasures, I'm almost human again. While most programs have provisions for keyboard control via the arrow keys, some games, especially flight simulator types, just work better with a joystick.

Which brings us to the A2D Joystick. Before obtaining the A2D unit, I had "borrowed" the sticks from my Color Computer. The advantage of the Color Computer sticks is that they are cheap, about $10 each. The disadvantages are that they break and bend. After a couple of good rounds they tend to develop an off-center wobble as the stalk bends where it is connected to the ball. These sticks also have only one button while certain applications require a two-button controller. They work, but a machine of the 1000's quality deserves better.

Answering the need for a more rugged and accurate joystick is Cinsoft, the distributor of the A2D Joystick. The control mechanism is not the traditional stick on a ball (bale stick assembly), but an open gimbal mechanism like those used to control radio-guided model aircraft. This mechanism is the standard for reliability and accuracy in R/C flying, therefore the control it offers for the joystick is excellent. Two switches on the bottom of the joystick give you the option of enabling or disabling either the horizontal or vertical spring-assisted centering. There are also two levers that allow adjusting of the zero centering point of both controls. The cord and connector are heavy duty and well-finished.

For those times when business activities get pushed onto the back burner, the A2D Joystick will put you in control of your recreational program.

(Cinsoft, 2235 Losantiville, Cincinnati, OH 45237, $29.95)

— Bruce Rothermel

Software **1000/1200/2000/3000**

A Complete Family Health History with *BDL Health*

BDL Health represents the consummate use of a home computer. It is a precursor of future computer use and is contained in a simple database.

BDL Health keeps track of the important facets of a family health program and requires only small and accurate input to do so.

Bette Laswell, author of the program, says it was born out of necessity for her family. The Laswells have moved around the country and Ms. Laswell was frustrated by the lack of medical records that followed their migration. Material supplied by physicians was sometimes more sketchy than a new doctor would want.

BDL Health works with 128K and one drive, but two drives or hard drive are very desirable. The program worked well on my 256K Tandy 1000. The menus and the instructions are very clear as is the documentation.

Ms. Laswell displays a great sense of humor in her programming. For example, if the user fails to make a choice among eight courses of action in the opening menu, he is met by the terse on-screen comment, "Mercy, Maude! We're just getting started and we're already arguing. I only know numbers here, specifically 1 to 8. Try again please." I found that much more comforting than the usual error message.

BDL Homeware's goal is to create software that is self-starting and self-instructing. *BDL Health* meets both these requirements. You are prompted through the program in an easygoing way, which should bolster the confidence of the neophyte.

There's room for 9,999 records. The records are kept for individual members of the family. Categories of health activities include: office visits, hospitalizations, other health occurrences, medications, tests, surgery, treatments and complete medical history. Entries are stored in a logical manner for future use. There are valuable functions such as the logging of prescriptions and other medicines, and possible reactions you may have had. These can be flagged so they can be retrieved in a search for allergic responses.

I was able to move among the various choices with ease. I found it easy to operate the program before reading all of the documentation. The documentation is as well-written as the program, so don't neglect it, but you really won't need it.

There is provision to produce a blank form to take with you to an office visit, so the needed data such as blood pressure, temperature and diagnosis can be written down. I wish I had such a record of past office visits!

With the proper information, this program can be a valuable tool to help organize information about your family's health, which can be of great value to you and your physician.

(BDL Homeware, 2509 N. Campbell Avenue, #328, Tucson, AZ 85719, 602-577-1435, $39.95)

— Howard Lee Ball

Software 100/200

C-Bug — A Good Debugger for the BASIC Programmer

C-Bug is a programming aid that can be of great help in debugging or modifying large BASIC programs. It is simple to use and does its job well. Normally when you are trying to debug a BASIC program, various PRINT statements have to be put in the program to see its progress. *C-Bug* lets you print out or display the contents of every variable with but a single CALL command. This is helpful because the CALL is easier to edit in than a PRINT statement with a long list of variables, and also because *C-Bug*'s thoroughness in listing every variable in use may show one you have overlooked.

The output of *C-Bug*, which lists the line number where called and the variable contents line by line, may be on paper or on the screen.

C-Bug is shipped on a noncopy-protected cassette with a user's guide. You start by running a simple BASIC program, *BUGLDR*, from tape; it leads you step by step through the process of loading the machine language program *C-Bug* into the Model 100. Queue Software provides the capability of relocating *C-Bug* to any desired region of memory, which allows it to coexist with other machine language programs.

BUGLDR gets its input from a second tape file called BUGASC.DO and produces executable code in RAM (670 bytes) as well as a CO file (C-BUG.CO, 677 bytes), if desired. *BUGLDR* is fast — it uses machine language as well as BASIC to quickly perform the relocation and checksum calculation. It uses some 1,500 bytes, but may be cleared from memory after RAM:C-BUG.CO has been created.

The four pages of documentation are clear, straightforward and thorough. Examples are given that illustrate every possible aspect of *C-Bug*.

I have only a few minor complaints with the documentation. It wouldn't hurt if the manual were to add that *C-Bug* goes wild with dimensioned variables, listing every possible element in the array. An array dimensioned as, say, DIM A(4,4,4) causes *C-Bug* to pour out 125 lines showing the values of all 125 elements, from A(0,0,0) to A(4,4,4). This may make *C-Bug* unusable for programs containing many arrays. Also, the BREAK key does not stop *C-Bug* during the 125-line printout.

Finally, where string variables have been used, the exact contents of the variable are sent unfiltered to the screen. Thus, if a string contains a CHR$(7) a beep is produced, while a CHR$(12) homes the cursor. Control and escape codes could wreak havoc on a printer whenever *C-Bug* is invoked.

No phone number is given in the manual, but a product as simple and straightforward as *C-Bug*, with its good documentation, does not really require telephone support.

If you do a lot of BASIC programming, *C-Bug* will be a big help. It is a good buy at $19.95.

(Queue Software Systems, 4528 Belleview, Suite 210, Kansas City, MO 64111, $19.95)

— Carl Oppedahl

Software 1000/1200/3000

Proteus — The Idea Processor

Proteus, first of all, is not a word processor. It is a program that offers the latest techniques in pre-writing. In fact, it forces you to use them. You could achieve the same results with any word processor, a freshman process text on composition and a lot of self-discipline.

Upon booting the program it invites you to name the current session, then gives you a main menu. Your writing choices are freewriting/looping, listing, five W's and cubing. From anywhere in the program you can use the up-arrow key to get to a "return to the main menu" prompt. *Proteus* is menu driven and easy to use once you get past the frustration of what it won't let you do.

Freewriting means just that — write as fast as you can; do not worry about spelling and punctuation — just get everything down on paper. (Remember, this is pre-writing. The actual writing comes later, when you rewrite.) *Proteus* helps you by typing everything in capital letters and by refusing to let you back up.

The program seems to miss the point of looping (which is to write for several short periods about the same idea and continue to loop back to the idea). All the other choices are what their names suggest. There is a help file that provides a simple explanation and an example of each option.

There is no way, within *Proteus* itself, to manipulate text. *Proteus* would be more beneficial if it had a few of the features of a basic word processor. You can, however, convert a *Proteus* session to an ASCII file for use with your own word processor. Then, instead of a file with text in uppercase, you have text in all lowercase, complete with typos, misspellings, outrageous punctuation and those initial awkward phrases. But you do have ideas, and they are saved on a disk. At this point, you are ready to begin writing.

Proteus will interface with existing word processors sold by Tandy, including *DeskMate*.

(Research Design Associates Inc., P.O. Box 848, Stony Brook, NY 11790, 516-928-5700, $59.95. Radio Shack Express Order System catalog number 90-598)

— Millard Dunn

Hardware

Pro-Tech-Tor — Electrical Outlet with a Plus

The Pro-Tech-Tor does just what its name implies. It's an excellent guard against electrical damage to your valuable electronic equipment. The outlet center, which has six grounded, 15-amp outlets, is equipped with a transient voltage surge supressor and a 125-volt, 15-amp circuit breaker.

The Pro-Tech-Tor has an LED grounded outlet indicator, and a lighted on/off switch that enables you to cut the power to your connected equipment without unplugging the suppressor or the equipment.

The center is housed in a "computer beige" epoxy finished aluminum case, has a six-foot cord, plugs directly into any 120-volt grounded outlet and has a two-year warranty.

(Northeast Peripherals, Inc., R.D. #1, Box 44, Somerset, NJ 08873, 800-526-2396, $24.95)

— Sue Rodgers

New Products

The following products have recently been received by PCM, examined by our magazine staff and approved for the PCM *Seal of Certification*, your assurance that we have seen the product and have ascertained that it is what it purports to be. This month the *Seal of Certification* has been issued to:

Ballyhoo, an interactive fiction game where you explore a circus lot while trying not to be caught by a lurking kidnapper. Includes program disk and circus paraphernalia. Requires Tandy 1000, 1200 or 3000. *Infocom, 125 Cambridge Park Drive, Cambridge, MA 02140, $39.95.*

The BASIC Handbook, Third Edition, encylopedia of the BASIC programming language by David A. Lein. Reference tool for converting between different dialects of BASIC. *CompuSoft Publishing, 535 Broadway, Dept. 12, El Cajon, CA 92021-5463, $24.95.*

CPYAT2PC, a copy utility that allows you to copy onto double-density disks with a high-density disk drive. Requires Tandy 3000. *Microbridge Computers International Inc., 655 Sky Way, Suite 125, San Carlos, CA 94070, (415)593-8777, $79.*

Digiware Graphics Utility, a graphics plotting utility designed for use with digitizers. Requires Tandy 1000, 1200 or 3000 and digitizer. *Computer Mathware, P.O. Box 1327, Princeton, NJ 80540, (609)924-6582, $300.*

DOS Helper, online reference system for MS-DOS. Requires Tandy 1000, 1200 or 3000. *Aristo Software, 16811 El Camino Real, Suite 213, Houston, TX 77058, (713)480-6288, $35.*

Filebase, simplified database management tool. Requires Tandy 1000, 1200 or 3000. *EWDP Software Inc., P.O. Box 40283, Indianapolis, IN 46240, (317)872-8799, $125.*

Forecast!, helps teach the science of meterology. Given current meterogical data, it helps predict the weather. Requires Tandy 1000, 1200 or 3000. *CBS Software Inc., Greenwich, CT 06863, $49.95.*

LapCoder, a text compression program for the Tandy 100 and 200. *Mu-Psi Computer Consulting, 1010 Turquoise Street, Suite 250, San Diego, CA 92109, (619)488-2350. $27.95 plus $2 S/H.*

LCdraw, a graphics editor for the Tandy 100. Requires Tandy 100 and 24K. *Lonestar Software and Integrated Circuit Design, 1611 Palmetto Lane, Kingwood, TX 77339, (800)835-2246, $49.95.*

Mapware Graphics Software, a program for creating professional graphics presentations incorporating maps. Requires Tandy 1000, 1200 or 3000 and plotter. *Computer Mathware, P.O. Box 1327, Princeton, NJ 80540, (609)924-6582, $500.*

PC LOGO, LOGO programming language and tutorial. Requires Tandy 1000, 1200 or 3000. *Harvard Associates, 260 Beacon Street, Somerville, MA 02143, $149.95.*

Plans 'n Totals III, V, VIII, a program for financial planning and forecasting. Requires Tandy 1000, 1200 or 3000. *Resource N Corporation, 66 Commonwealth Avenue, Concord, MA 01742, (617)264-4450, III $195; V $395; VIII $695.*

Q-DOS, a hard disk files management utility. Requires Tandy 1000, 1200 or 3000 and hard disk drive. *Gazelle Systems, 230 N. 2475 W., Provo, UT 84601, (810)377-1288, $29.95.*

The Recipe Writer, maintains thousands of recipes which can be cross-referenced into any chosen categories. Requires Tandy 1000, 1200 or 3000. *At-Your-Service Software Inc., 309 W 97th Street, New York, NY 10025, (212)866-2186, $79.*

By awarding a *Seal*, the magazine certifies the program *does exist*, but this *does not* constitute any guarantee of satisfaction. As soon as possible, these hardware or software items will be forwarded to PCM's reviewers for evaluation.

SUPER SAVINGS ON TANDY 1000 UPGRADES

SAVE!!

★ 640K Memory Upgrade Board with Real Time Clock .... **Only $189.95**
★ Serial Interface Board ..... **Only 49.95**
★ Parallel Interface Board .... **Only 29.95**
★ Hayes Compatible 300/1200 Baud Deluxe Modem Board .... **Only 189.95**

ACCESSORIES

Logitech Mouse - uses standard serial port .... **Only $ 99.95**
Logitech Mouse with PC Paint program **129.95**
Monitors - Zenith, CGS, Taxan Call for details ..... **From 99.95**
Star SG-10 printer **249.95**
Teac, Panasonic D.S.D.D. drives **119.95**
10 discs in an attractive styrene case **12.95**
Deluxe keyboard drawer 3½" high computer platform with sliding keyboard drawer. Ball bearing glides and locks. Enamel finish to match your PC **69.95**

IBM COMPATIBLE PCX-II

IBM Compatibility at sensational prices!!
Starting at only $685.00

Complete, deluxe 640K system includes 150 watt power supply, 2 Teac floppies, printer and serial port, clock, amber monitor and many other extras.

$1080.00
20 meg hard drive - $525.00

We maintain a library of quality public domain software for purchasers of our computer systems — text processor, data base manager, spread sheet, communication, utilities and entertainment software. Call for details.

MS-DOS SOFTWARE

DAC ACCOUNTING - General ledger, receivables, payables, inventory. **Unbelievable bargain at only .. $69.95**
MARK DATA - Our popular ledger system with detailed operating manual. ..... **Only $69.95**
DBASE III - Ashton Tate's powerful database management system. **$389.00**
The Twin - Works exactly like Lotus 1-2-3 Outstanding value. **$145.00**
VP Planner - Another great Lotus look-alike. ..... **Only $99.95**
CoCo Util II - A valuable utility program that allows you to transfer Tandy CoCo disc files to your MS-DOS machine. You may also transfer MS-DOS files to a Color Computer disc. Save countless hours of typing. ..... **New Version $39.95**
or upgrade from Version I $12.95
including shipping.

MD Mark Data Products

24001 Alicia Parkway, #207
Mission Viejo, CA 92691 (714) 768-1551

SHIPPING: All orders under $100 please add $2 regular, $5 air. All orders over $100 please add 3% regular, 8% air. California residents please add 6% sales tax. Orders outside the continental U.S., check with us for shipping amount; please remit U.S. funds. Software authors — contact us for exciting program marketing details. We accept MasterCard and VISA. Distributed in Canada by Kelly Software.

Using BAREAD 2.1

Bar code listings must be read in numerical order beginning with Line 1 and continuing through the last line of the listing. The computer display is used to prompt you as to which line to scan and give you warning messages should you happen to get out of step.

When you run *BAREAD*, it asks you to scan the first line of the bar code listing. This line contains the name of the program as well as the beginning of the program itself. The computer will sound a high-pitched beep whenever it's ready for you to scan a line. After a line has been successfully read, you'll hear a lower beep. A "blip-bloop" sound prompts you to turn your attention to the screen for a message. You'll hear this when you accidentally scan a line out of sequence.

BAREAD 2.1

```
1ØØØ ' *** Initialize ***
1Ø1Ø ON ERROR GOTO 1Ø4Ø
1Ø2Ø CLEAR 1ØØØ:MAXFILES=2
1Ø3Ø GOTO 1Ø5Ø
1Ø4Ø IF ERR=5 THEN RESUME NEXT
1Ø5Ø ON ERROR GOTO Ø
1Ø6Ø RUNM "B3OF9"
1Ø7Ø OPEN "WAND:" FOR INPUT AS #1
1Ø8Ø UC%=-1
1Ø9Ø PC$="Ø123456789ABCDEFGHIJKLMNOPQRST
UVWXYZabcdefghijklmnopqrstuvwxyz- $+"
11ØØ DIM RW$(36)
111Ø ER$(1)="You must scan line 1 first!
"
112Ø ER$(2)="You've SKIPPED a line!"
113Ø ER$(3)="You've ALREADY SCANNED this
 line!"
114Ø ER$(4)="Code not PCM2/39 format!"
115Ø ER$(5)="Command not applicable here
!"
116Ø ER$(6)="You cannot skip this line!"
117Ø ER$(7)="Selected resume file not in
 computer!"
118Ø ' *** Read Reserved Words List ***
119Ø DATA BEEP,CLEAR,CLOSE,DATA,DEFDBL,D
EFINT,DEFNG,DEFSTR,ELSE,GOSUB,GOTO
12ØØ DATA INKEY$,INPUT,INSTR(,LCOPY,LEFT
$(,LINE(,LOADM,LPRINT,USING,MAXFILES
121Ø DATA MID$(,NEXT,PEEK,POKE,POWER,PRE
SET(,PRINT,READ,RESTORE,RETURN,RIGHT$(
122Ø DATA SOUND,SPACE$(,STRING$(,THEN
123Ø FOR I%=1 TO 36:READ RW$(I%):NEXT I%
124Ø ' *** Procedure Begins Here ***
125Ø CLS:PRINT@44,"PCM Bar Code Program
Reader v2.1"
126Ø LINE(2Ø,4)-(219,18),1,B:LINE(22,6)-
(217,16),1,B
127Ø NN%=1
128Ø GOSUB 166Ø:IF ER%>Ø THEN GOSUB 162Ø
:GOTO 128Ø
129Ø IF LL%=Ø AND INSTR("YN",IL$)>Ø THEN
 ER%=5:GOSUB 162Ø:GOTO 128Ø
13ØØ IF LL%=Ø THEN ON INSTR("ALSR",IL$)
```

After reading the first line, you continue scanning with the second line. Remember to wait for a high beep before scanning and then listen for a low beep to indicate a successful read.

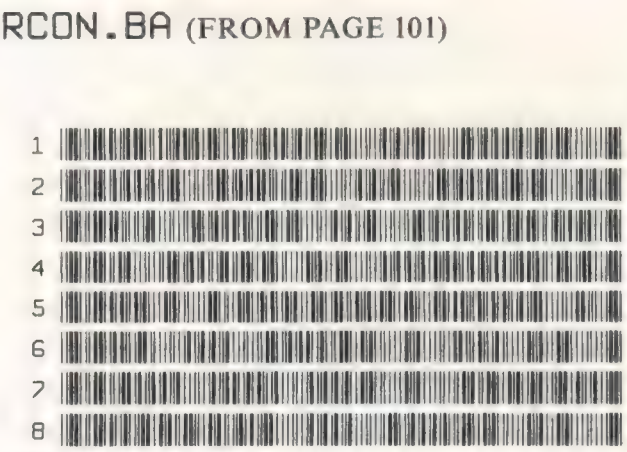
Once the last line of the listing has been scanned, *BAREAD* will return control to the Tandy 100/200 menu screen. Note that the program you just scanned is now in the directory with a .DO extension.

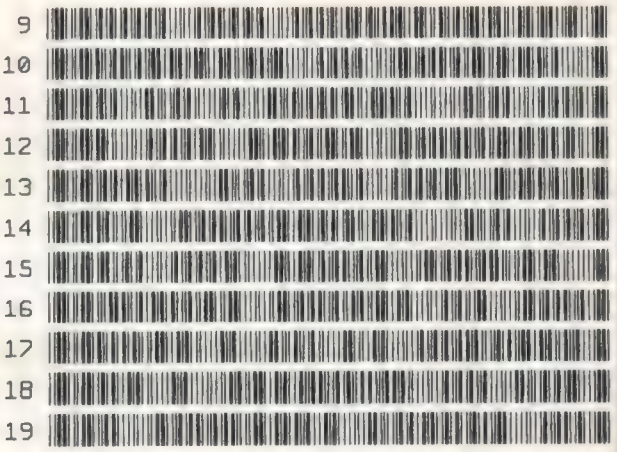
The final step is to convert the .DO text file to a normal BASIC program. This is done quite simply by going to BASIC and loading the file with a command such as LOAD"TEST .DO" (if the program name were TEST). The program will load into BASIC and will be ready to run. To save the program in BASIC's compressed format (.BA extension), you'd type SAVE"TEST" (if the program were named TEST). You may then kill the .DO file with KILL "TEST.DO".

```
GOTO 182Ø,189Ø,198Ø,2Ø5Ø
131Ø IF LL%=1295 THEN 135Ø
132Ø IF LL%<>NN% AND NN%=1 THEN ER%=1:GO
SUB 162Ø:GOTO 128Ø
133Ø IF LL%<NN% THEN ER%=3:GOSUB 162Ø:GO
TO 128Ø
134Ø IF LL%>NN% AND NN%>1 THEN ER%=2:GOS
UB 162Ø:GOTO 128Ø
135Ø IL$=RIGHT$(IL$,19)
136Ø IF LL%=1 AND NN%>Ø THEN GOSUB 178Ø
137Ø CL$=CL$+IL$
138Ø FOR I%=1 TO LEN(CL$)
139Ø    CH$=MID$(CL$,I%,1)
14ØØ    IF CH$="%" THEN GOSUB 151Ø:IF NL
% THEN 147Ø ELSE GOTO 144Ø
141Ø    IF CH$="/" THEN GOSUB 155Ø:IF NL
% THEN 147Ø ELSE GOTO 144Ø
142Ø    IF CH$="." THEN UC%=NOT(UC%):GOT
O 145Ø
143Ø    IF CH$=>"A" AND CH$<="Z" AND NOT
(UC%) THEN CH$=CHR$(ASC(CH$)+32)
144Ø    XX$=XX$+CH$:IF RIGHT$(XX$,1)=CHR
$(13) THEN PRINT#2,XX$;:XX$="":UC%=-1
145Ø NEXT I%
146Ø CL$=""
147Ø PRINT@2ØØ,SPACE$(8Ø);
148Ø IF LL%<>1295 THEN NN%=LL%+1:GOTO 12
8Ø
149Ø ' *** Done ***
15ØØ CLOSE:CALL 618Ø7!:CLEAR 5ØØ,HIMEM:M
ENU
151Ø ' *** Decode Reserved Word ***
152Ø NL%=Ø:IF I%>LEN(CL$)-1 THEN NL%=-1:
CL$="%":GOTO 154Ø
153Ø I%=I%+1:CH$=RW$(INSTR(PC$,MID$(CL$,
I%,1)))
154Ø RETURN
155Ø ' *** Decode Hex and Control Charac
ters ***
156Ø NL%=Ø:IF I%>LEN(CL$)-1 THEN NL%=-1:
CL$="/":GOTO 161Ø
157Ø I%=I%+1:IF INSTR("/%.",MID$(CL$,I%,
1))>Ø THEN CH$=MID$(CL$,I%,1):GOTO 161Ø
158Ø IF I%>LEN(CL$)-1 THEN NL%=-1:CL$=RI
```

```
GHT$(CL$,2):GOTO 161Ø
159Ø HX$=MID$(CL$,I%,2):CH$=CHR$((INSTR(
"Ø123456789ABCDEF",LEFT$(HX$,1))-1)*16+I
NSTR("Ø123456789ABCDEF",RIGHT$(HX$,1))-1
)
16ØØ I%=I%+1
161Ø RETURN
162Ø ' *** Error Codes ***
163Ø SOUND 5ØØØ,1Ø:SOUND 8ØØØ,1Ø:SOUND 5
ØØØ,1Ø
164Ø PRINT@22Ø-.5*LEN(ER$(ER%)),ER$(ER%)
;
165Ø RETURN
166Ø ' *** Get Code Line ***
167Ø PRINT@173,"";:PRINT USING "Scan lin
e ###";NN%
168Ø IF NN%=-1 THEN PRINT@173,"Scan any
line":GOTO 17ØØ
169Ø SOUND 5ØØ,5
17ØØ INPUT#1,IL$:ER%=Ø
171Ø FOR I%=1 TO LEN(IL$)
172Ø IF MID$(IL$,I%,1)="!" THEN MID$(IL$
,I%,1)="."
173Ø NEXT I%
174Ø IF LEN(IL$)<>1 AND LEN(IL$)<>21 THE
N ER%=4:RETURN
175Ø IF LEN(IL$)=1 THEN LL%=Ø:RETURN
176Ø LL$=LEFT$(IL$,2):LL%=(INSTR("Ø12345
6789ABCDEFGHIJKLMNOPQRSTUVWXYZ",LEFT$(LL
$,1))-1)*36+INSTR("Ø123456789ABCDEFGHIJK
LMNOPQRSTUVWXYZ",RIGHT$(LL$,1))-1
177Ø RETURN
178Ø ' *** Open Program File ***
179Ø PN$=LEFT$(IL$,6):IL$=RIGHT$(IL$,LEN
(IL$)-6)
18ØØ OPEN PN$ FOR OUTPUT AS #2
181Ø RETURN
182Ø ' *** Abort ***
183Ø BEEP:BEEP:BEEP
184Ø PRINT@2Ø9,"ABORT!  Are you sure?";
185Ø INPUT#1,AN$
186Ø IF INSTR("YN",AN$)=Ø THEN BEEP:PRIN
T@251,"Scan 'YES' or 'NO'":GOTO 185Ø
187Ø PRINT@2ØØ,SPACE$(8Ø);
188Ø IF AN$="Y" THEN CLOSE:KILL PN$+".DO
":GOTO 149Ø ELSE GOTO 128Ø
189Ø ' *** Skip Line ***
19ØØ IF NN%=1 THEN ER%=6:GOSUB 162Ø:GOTO
 128Ø
191Ø BEEP:BEEP:BEEP
192Ø PRINT@21Ø,"SKIP!  Are you sure?"
193Ø INPUT#1,AN$
194Ø IF INSTR("YN",AN$)=Ø THEN BEEP:PRIN
T@251,"Scan 'YES' or 'NO'":GOTO 193Ø
195Ø PRINT@2ØØ,SPACE$(8Ø);
196Ø IF AN$="Y" THEN NN%=NN%+1
197Ø GOTO 128Ø
198Ø ' *** Stop & Save ***
199Ø BEEP:BEEP:BEEP
2ØØØ PRINT@2Ø7,"STOP & SAVE! Are you sur
e?";
2Ø1Ø INPUT#1,AN$
2Ø2Ø IF INSTR("YN",AN$)=Ø THEN BEEP:PRIN
T@251,"Scan 'YES' or 'NO'":GOTO 2Ø1Ø
2Ø3Ø PRINT@2ØØ,SPACE$(8Ø);
2Ø4Ø IF AN$="Y" THEN 149Ø ELSE GOTO 128Ø
2Ø5Ø ' *** Resume ***
2Ø6Ø IF NN%<>1 THEN ER%=5:GOSUB 162Ø:GOT
O 128Ø
2Ø7Ø PRINT@254,"Resume Mode";
2Ø8Ø NN%=1:GOSUB 166Ø
2Ø9Ø IF LL%=Ø THEN ER%=5 ELSE IF LL%<>1
THEN ER%=1
21ØØ IF ER%>Ø THEN GOSUB 162Ø:GOTO 2Ø6Ø
211Ø PN$=MID$(IL$,3,6)
212Ø ON ERROR GOTO 214Ø
213Ø OPEN PN$ FOR INPUT AS #2:GOTO 217Ø
214Ø RESUME 215Ø
215Ø CLOSE #2
216Ø ER%=7:GOSUB 162Ø:GOTO 127Ø
217Ø CLOSE #2:OPEN PN$ FOR APPEND AS #2
218Ø NN%=-1:GOTO 128Ø
```

CURCON.BA (FROM PAGE 101)





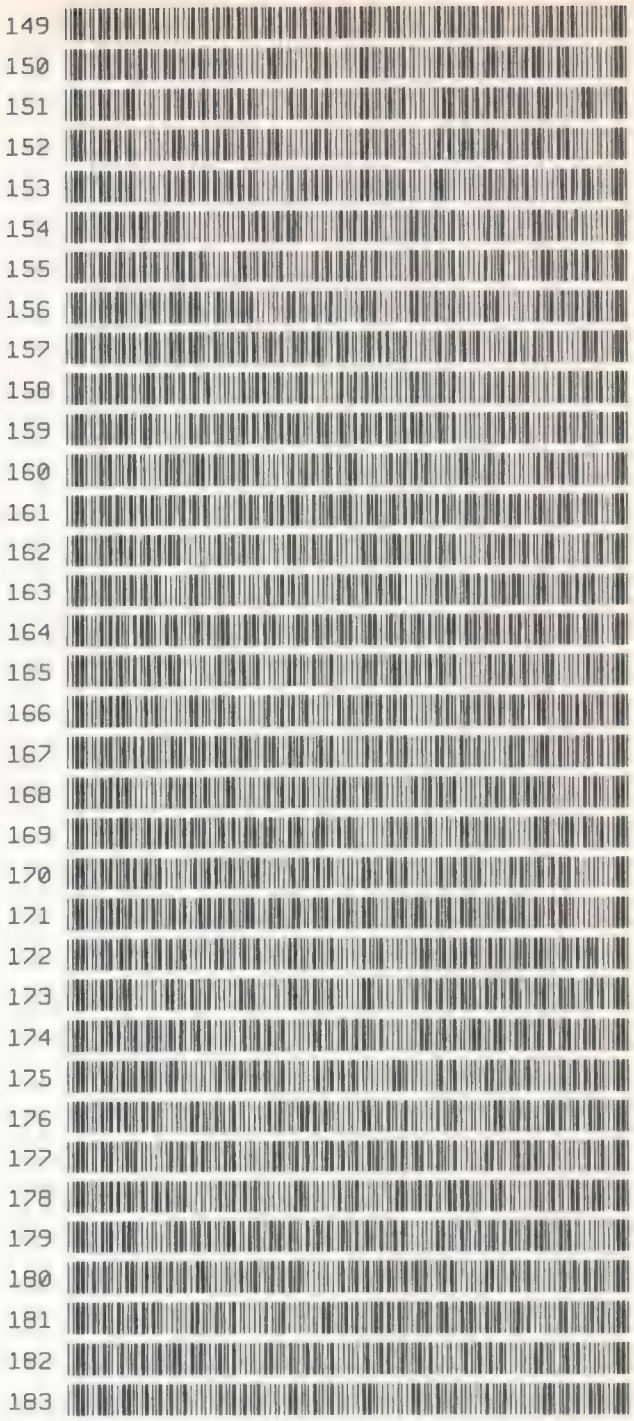
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

DACO.BA (FROM PAGE 47)

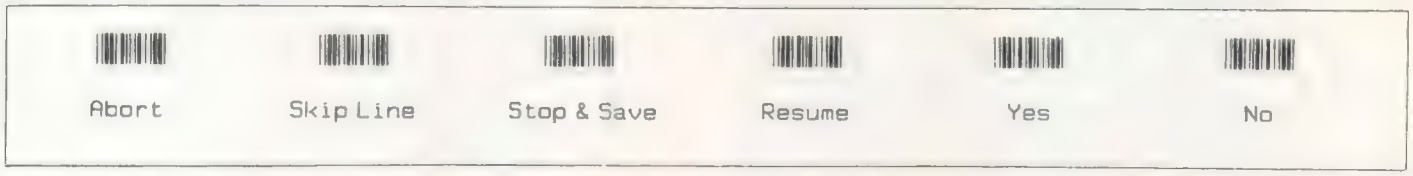
1
2
3
4
5
6
7
8
9

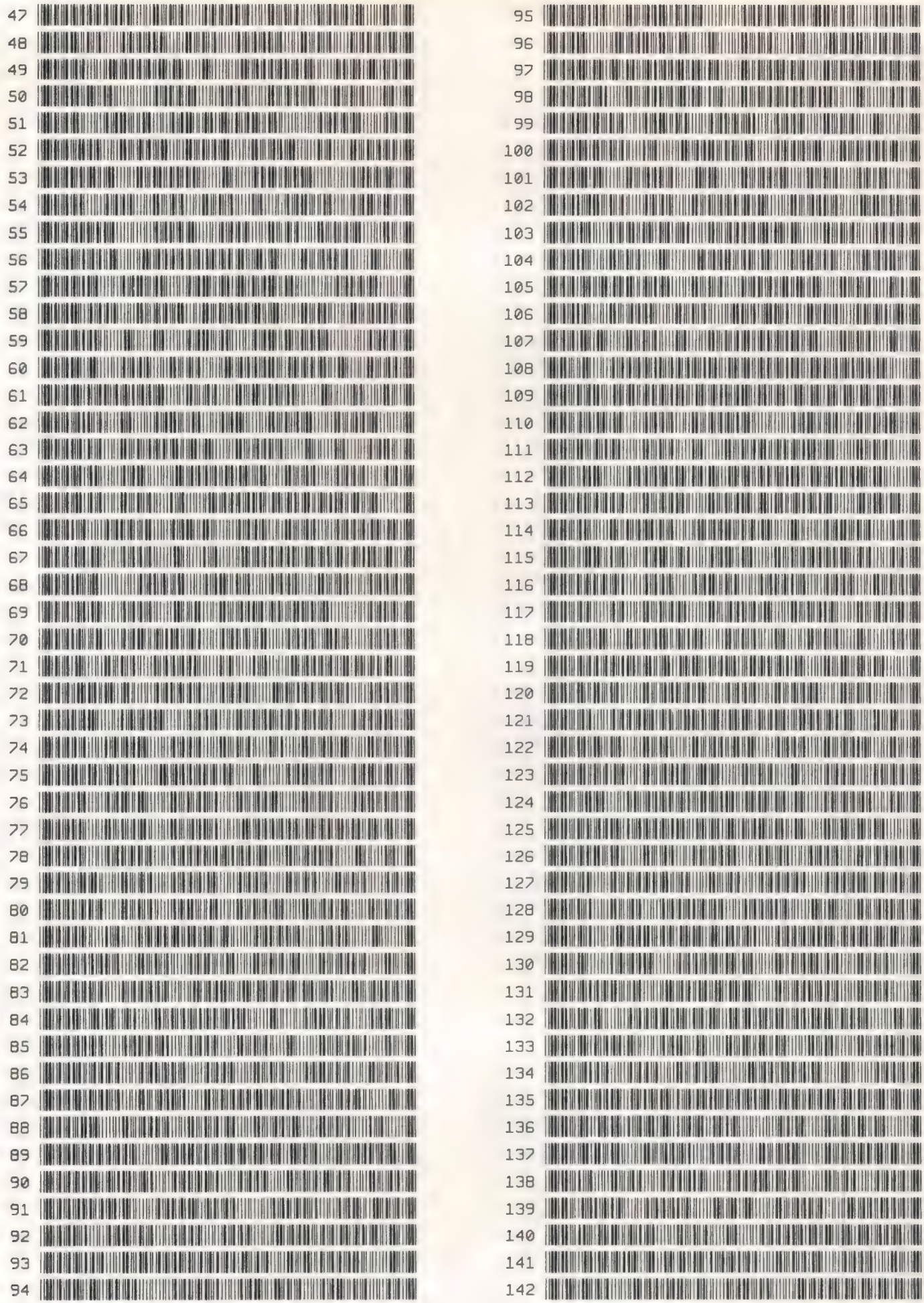
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

Abort	Skip Line	Stop & Save	Resume	Yes	No



DATMG.BA (FROM PAGE 47)





Racksellers

ALABAMA	
Birmingham	Jefferson News Co.
Florence	Anderson News Co.
Greenville	M&B Electronics/Radio Shack
Madison	Madison Books
ARKANSAS	
Fayetville	Vaughn Electronics/Radio Shack
ARIZONA	
Phoenix	Computer Pro
	Tri-Tek
Sierra Vista	Livingston's Books
Tempe	Computer Library
Tucson	Anderson News Co.
CALIFORNIA	
Citrus Heights	Software Plus
Elk Grove	Micro Mainframe
Half Moon Bay	Strawflower Electronics
Livermore	Software Galeria
North Hollywood	Levity Distributors
Santa Rosa	Sawyer's News
Sunnyvale	Computer Literacy
COLORADO	
Salida	Radio Shack
DELAWARE	
Middletown	Delmar Co.
Wilmington	Normar Inc.—The Smoke Shop
FLORIDA	
Boca Raton	Software, Software, Inc.
Davie	Software Plus More
Ft. Lauderdale	Software Connection
N. Miami Beach	Almar Bookstore
Orlando	Book Mania
Panama City	Computer Systems Group
Pensacola	Anderson News Co.
Tallahassee	Anderson News Co.
GEORGIA	
Atlanta	Guild News Agency
ILLINOIS	
Chicago	Chas. Levy Distr.
	Kroch's & Brentano's Jackson Street
	Kroch's & Brentano's Wabash Ave.
	Prairie News Agency
Oakbrook	Kroch's & Brentano's
INDIANA	
Highland	Computer Health Consultants
Scottsburg	Radio Shack of Scottsburg
IOWA	
Ames	Myers TV & Stereo Repair
KANSAS	
Wichita	Lloyd's Radio
KENTUCKY	
Georgetown	Goodwin Electronics
Louisville	Hawley-Cooke Bookstores
Prospect	Falsoft, Inc.
LOUISIANA	
Crowley	Acadiana News
Gretna	Computer Supply Store
Slidell	Radio Shack
MASSACHUSETTS	
Littleton	Computer Plus
Marlboro	Radio Shack
MICHIGAN	
Lowell	Curt's Home & Sound Arcade
Muskegon	The 8-Bit Corner
Rochester	Rochester Book Center
Sterling Heights	Programs, Inc.
	Software City
Wyoming	Gerry's Book Co.
MISSOURI	
Independence	Digital Enterprises, Inc.
St. Louis	Software Center
NEBRASKA	
Lincoln	Hobby Town
NEW HAMPSHIRE	
West Lebanon	Verham News Corp.
NEW JERSEY	
Clinton	Micro World II
Hackensack	Total Circulation Service
Pennsville	Dave's Electronic Radio Shack
NEW MEXICO	
Alamogardo	New Horizons Computer Systems
Albuquerque	News and Bookstore
	Page One Newsstand
NEW YORK	
Fredonia	On Line: Computer Access Center
Hudson Falls	G.A. West & Co.
NORTH CAROLINA	
Cary	News Center in Cary Village
Sylva	Total Data Systems
NORTH DAKOTA	
Fargo	Computer Associates
OHIO	
Blanchester	J R Computer Control
Cincinnati	Cinsoft
Dayton	Huber Heights Book & Card
Girard	Girard Book & News
Kenton	T.W. Hogan & Assoc., Inc.
Mount Orab	Mount Orab Radio Shack
Toledo	Leo's
OKLAHOMA	
Oklahoma City	Merit Computers
OREGON	
Portland	Fifth Avenue News
Woodburn	Pioneer Electronics
PENNSYLVANIA	
Philadelphia	Newsy
Phoenixville	Stevens' Radio Shack
Wind Gap	Micro World
RHODE ISLAND	
Newport	Kelly's Variety
TENNESSEE	
Knoxville	Anderson News Co.
Memphis	Computer Center
Nashville	Misko's Book Store
Smyrna	Delker Electronics, Inc.
TEXAS	
Bay City	Impulse Computers
Ft. Worth	A & A International
VIRGINIA	
Alexandria	Alonso Book & Period
Norfolk	I-O Computer
Wytheville	Wytheville Electronics
WASHINGTON	
Kennewick	Satellite TV Specialists
WEST VIRGINIA	
Lesage	Pioneer Technology
WISCONSIN	
Ladysmith	Electronics, Etc.
Madison	20th Century Books
ARGENTINA	
Cordoba	Informatica Y Telecomunicaciones
CANADA:	
ALBERTA	
Calgary	Rainbow Software Services
Edmonton	CMD Micro
	Kelly Software Distributors
BRITISH COLUMBIA	
Sidney	Sidney Electronics
NOVA SCOTIA	
Halifax	Atlantic News
QUEBEC	
LaSalle	Messageries De Presse Benjamin

Also available at these selected bookstores: B. Dalton, Pickwick Books, Encore Books, Barnes & Noble, Little Professors, Tower Book & Records, Kroch's & Brentano's, and Community Newscenters.

Advertisers Index

PCM Advertising and Marketing Representatives

Shackleford, Nolan, Davis, Gregg and Associates
Cindy J. Shackleford, president
12110 Meridian South, Suite 5
P.O. Box 73-578
Puyallup, WA 98373-0578
(206) 848-7766

Kim Vincent
Advertising Representative
The Falsoft Building
9509 U.S. Highway 42
P.O. Box 385
Prospect, KY 40059
(502) 228-4492

TANDY 1000

Howard Medical Computers

*pbj Multifunction Board

$276

List $359.95
MFB-1000
with 256K $269
with 128K $249

- 512 K RAM included
- Serial Port for modem or printer
- Battery powered real time clock/calendar
- RAM disk program included
- Acts like hard drive only faster

*pbj X Ram Card with 256K

$249

List $299.95
Add $49 for each
Bank of 256K

- Add additional memory to your 1000 up to 2 megabytes
- Breaks the 640K barrier
- Includes diagnostic package & RAM disk program

*Zenith Green Monitor

$67.50

Was $149
$7 shipping

- 12" Zenith 123A Green Screen is easy on the eyes
- High resolution: 640 dots x 200 dots
- 15 MHz band width • non glare screen
- Composite video monochrome
- 30 day warranty at over 1200 Zenith repair locations

*Epson LX-80 Printer

REPACK 80 COLUMN

$240

$7 shipping

- Near letter quality or 100 cps draft mode
- Friction feed standard ($30 tractor option)
- Bidirectional with 1K graphics buffer

Payrol/BAS™

$149

Complete
$7 shipping

- No tables to enter. Automatically calculates Federal, FICA, State and up to 5 individual deductions
- Handles 500 employees with hard drive or (40) employees with one floppy drive plan weekly
- User friendly, easy to install
- Prints checks, 941 summary and W-2's

Send for FREE Reports Guide
or $5 for Demo Disc

1690 Elston, Chicago, IL 60622 (800) 443-1444 orders (312) 278-1440 questions

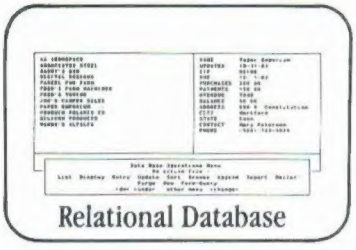
*Howard Medical offers a 30-day return guarantee on all hardware we sell. Return equipment to us within 30 days for a refund if you are unsatisfied for any reason.

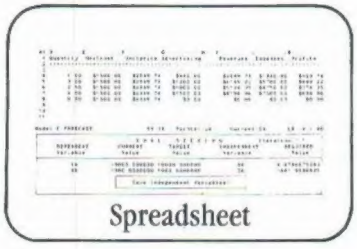
Top-of-the-line: *Open Access*®

NOW ONLY $395*

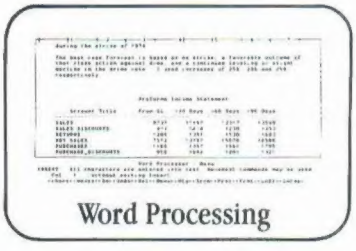
OPEN ACCESS Comparison Chart	Database	Form Query	Report Generator	Query Processor	Relational Database	Graphics	3-D Graphics	Spreadsheet	Goal Seeking	Word Processor	Communications	Time Management	Price
Lotus 1-2-3	✓					✓		✓					$495
Symphony	✓	✓	✓			✓		✓		✓	✓		695
Framework	✓		✓	✓		✓		✓		✓	✓		695
Smart System	✓		✓		✓	✓		✓		✓	✓	✓	895
Open Access	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	**395***

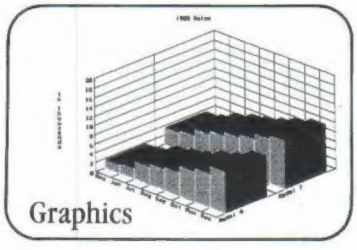
Open Access has sold almost 50,000 copies overseas, and many more in the U.S. thanks to Radio Shack®. With one of the most powerful Relational Databases around, and just about all the additional features you could want, it's no wonder. Just the Goal Seeking feature in the Spreadsheet alone should make *Open Access* your customer's #1 choice.
(We're even written in 11 languages with exact file transfer.)

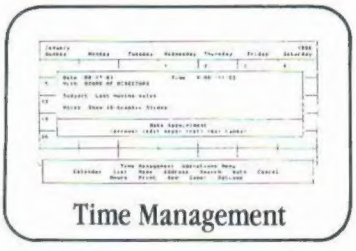

Relational Database


Spreadsheet


Communications


Word Processing


Graphics


Time Management

Available through your exclusive *Express Order Software System*

90-0114 Tandy2000 • 90-0247 Tandy1200 • 90-0437 Tandy1000

The product's exceptional. The price extraordinary. Can you really afford to compromise any longer?

SPI

SOFTWARE PRODUCTS INTERNATIONAL

10240 Sorrento Valley Road ■ San Diego, CA 92121

*U.S. version only

Open Access and ▣ are registered trademarks of Software Products International, Inc; Lotus 1-2-3 and Symphony are registered trademarks of Lotus Development Corp.; Framework is a trademark of Ashton-Tate; Smart System is a registered trademark of Innovative Software; Tandy 1000/1200/2000 are trademarks of Radio Shack, a division of Tandy Corp.

SUPER ROM

Lucid Spreadsheet Write ROM Database Outliner

"The Tandy Portable Drive with one button boot!"

NOW AVAILABLE ON THE M-200

NOW YOU CAN REALLY HAVE IT ALL!

All on one ROM. Truly the finest four programs available for the Model 100 — guaranteed. Try it for 30 days. If you aren't blown away by the excellence return it for a full refund.

$199.95

The four best programs for the Model 100 all on one ROM. 32K of power without using any RAM for program storage. This is the PCSG Snap-In ROM that just presses easily into the little ROM socket in the compartment on the back. You access the four right from the main menu like built-ins.

Write ROM — the definitive word processor for the Model 100. Function key formatting or dot commands. Search and replace. Library feature — inserts words, phrases or whole documents into text from just a code. MAP lets you see a picture of your document. In all there are 60 features and functions. No one can claim faster operation. FORM lets you create interactive forms with on-screen prompts that you can answer from the keyboard. Nothing else for the Model 100 compares with the features of Write ROM. Exactly the same as the Write ROM sold as a single program. Infoworld says it "makes the Model 100 a viable writing unit ... surpassed our highest expectations for quality and clarity."

Lucid Spreadsheet: This is the one PICO magazine says "blows Multiplan right out of the socket" and Infoworld performance rated as "excellent" and said "makes the Model 100 compute." Gives you features you cannot get with Lotus 123. Lets you build spreadsheets in your Model 100 that would consume 140-150K on a desktop. Program generating capability with no programming knowledge required. Variable column widths. Includes find and sort with function key control. It's fast, recalculates like lightning. No feature has been taken from the original, only new ones added.

Database: This is a relational data base like no other. You can do everything from mailing lists to invoices. No complicated pseudo-coding, you create input screens as simply as typing into TEXT. You are not limited by size; you can have as large an input screen as you wish. Prints out reports or forms, getting information from as many files as you like. Complete math between fields. Total interface with Lucid worksheets.

Outliner: Does everything that Thinktank does on a PC but a whole lot better. Includes a Sort for your headlines. Lets you have headlines of up to 240 characters. Has cloning, hoisting and sideways scroll up to 250 characters. Like Lucid, this one sets a new standard for outliners. This is the way to plan and organize your projects.

Present Lucid and Write ROM owners can upgrade for $150. If you have both it's $125.

As usual PCSG sells the Super ROM on a thirty day guarantee. If for any reason you are not satisfied, simply return it for a full refund.

We are excited about this product. Super ROM gives the Model 100 the true power of a desktop. No other multi-program ROM has software that compares. But don't take our word for it. We invite you to make that comparison yourself. Priced at $199.95 on Snap-In ROM.

Got stuck with somebody else's multi-ROM? We'll upgrade it for $150.

(214) 351-0564

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd., #206, Dallas, TX 75229

MC, Visa, American Express, Check, or C.O.D.

© PCSG